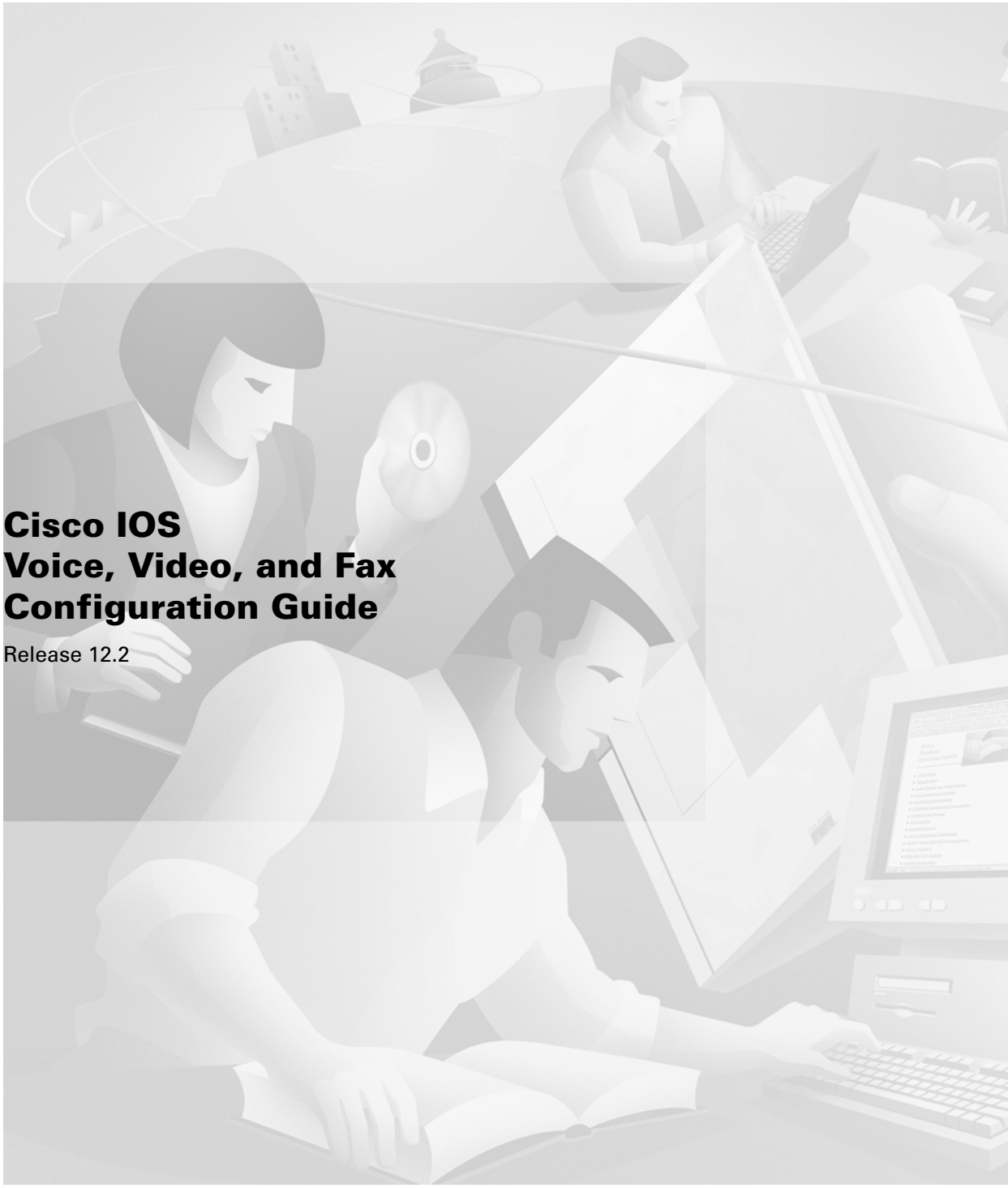
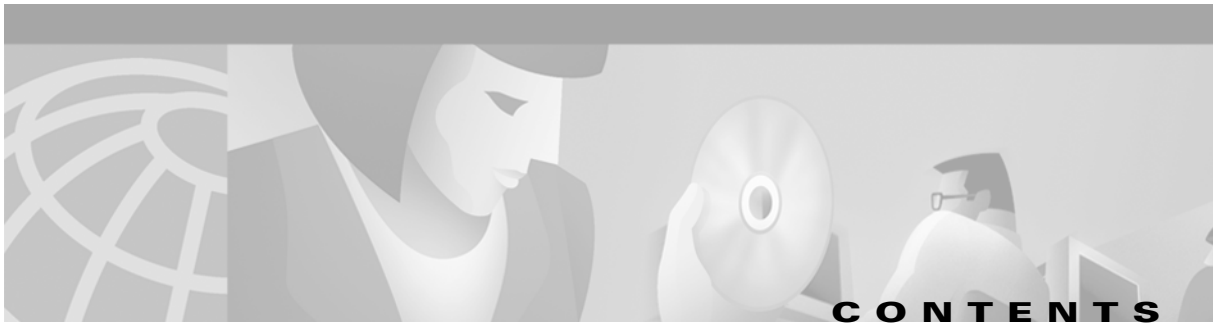


Cisco IOS Voice, Video, and Fax Configuration Guide

Release 12.2

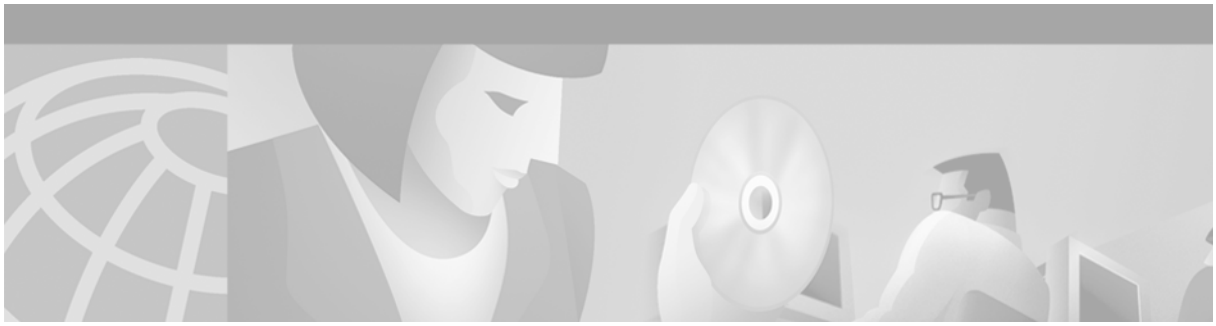
Corporate Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 526-4100

Customer Order Number: DOC-7812100=
Text Part Number: 78-12100-01



CONTENTS

Appendixes



About Cisco IOS Software Documentation

This chapter discusses the objectives, audience, organization, and conventions of Cisco IOS software documentation. It also provides sources for obtaining documentation from Cisco Systems.

Documentation Objectives

Cisco IOS software documentation describes the tasks and commands necessary to configure and maintain Cisco networking devices.

Audience

The Cisco IOS software documentation set is intended primarily for users who configure and maintain Cisco networking devices (such as routers and switches) but who may not be familiar with the tasks, the relationship between tasks, or the Cisco IOS software commands necessary to perform particular tasks. The Cisco IOS software documentation set is also intended for those users experienced with Cisco IOS software who need to know about new features, new configuration options, and new software characteristics in the current Cisco IOS software release.

Documentation Organization

The Cisco IOS software documentation set consists of documentation modules and master indexes. In addition to the main documentation set, there are supporting documents and resources.

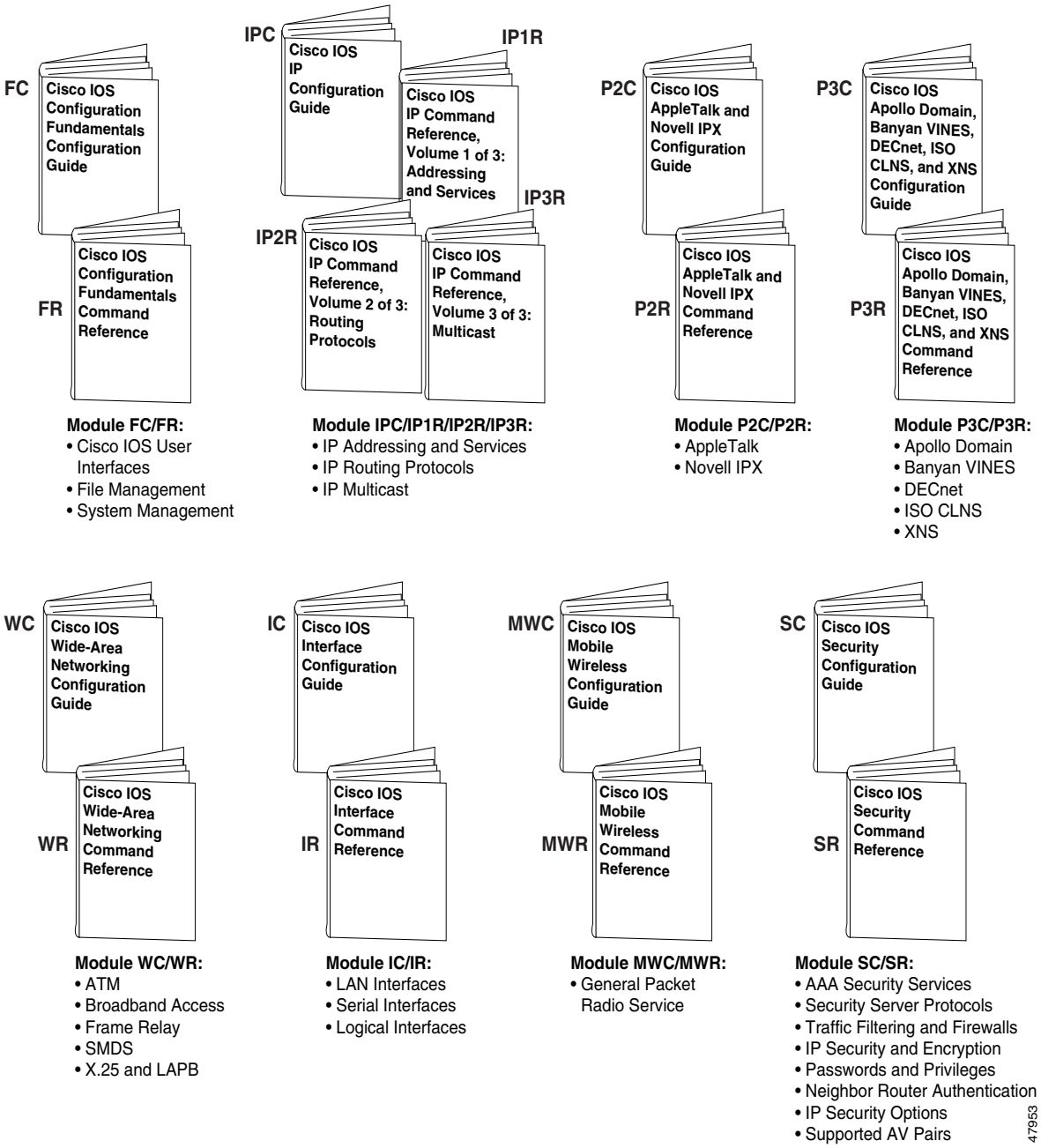
Documentation Modules

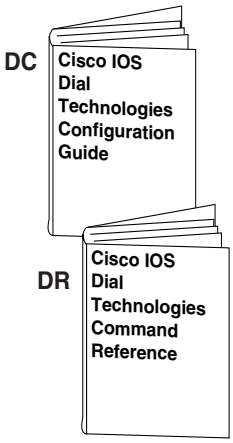
The Cisco IOS documentation modules consist of configuration guides and corresponding command reference publications. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command reference publication provide complete Cisco IOS command syntax information. Use each configuration guide in conjunction with its corresponding command reference publication.

Figure 1 shows the Cisco IOS software documentation modules.

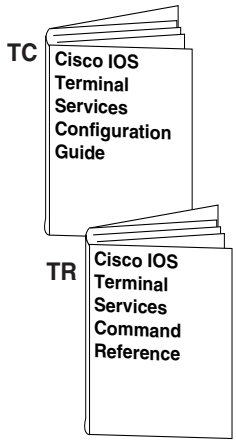
Note The abbreviations (for example, FC and FR) next to the book icons are page designators, which are defined in a key in the index of each document to help you with navigation. The bullets under each module list the major technology areas discussed in the corresponding books.

Figure 1 Cisco IOS Software Documentation Modules

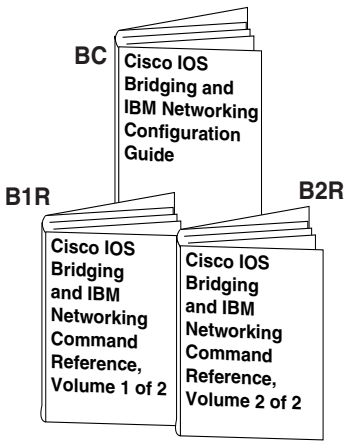




Module DC/DR:
- Preparing for Dial Access
- Modem and Dial Shelf Configuration and Management
- ISDN Configuration
- Signalling Configuration
- Dial-on-Demand Routing Configuration
- Dial-Backup Configuration
- Dial-Related Addressing Services
- Virtual Templates, Profiles, and Networks
- PPP Configuration
- Callback and Bandwidth Allocation Configuration
- Dial Access Specialized Features
- Dial Access Scenarios

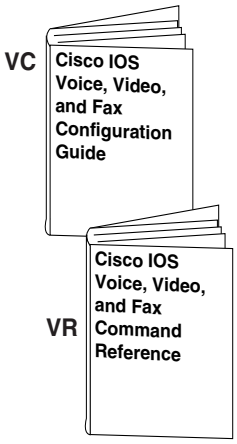


Module TC/TR:
- ARA
- LAT
- NASI
- Telnet
- TN3270
- XRemote
- X.28 PAD
- Protocol Translation

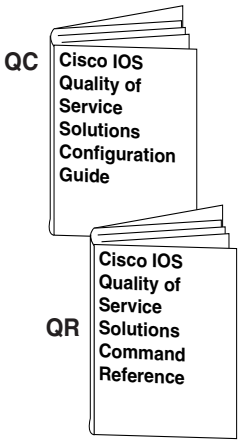


Module BC/B1R:
- Transparent Bridging
- SRB
- Token Ring Inter-Switch Link
- Token Ring Route Switch Module
- RSRB
- DLSw+
- Serial Tunnel and Block Serial Tunnel
- LLC2 and SDLC
- IBM Network Media Translation
- SNA Frame Relay Access
- NCIA Client/Server
- Airline Product Set

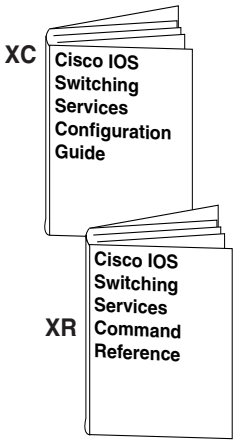
Module BC/B2R:
- DSPU and SNA Service Point
- SNA Switching Services
- Cisco Transaction Connection
- Cisco Mainframe Channel Connection
- CLAW and TCP/IP Offload
- CSNA, CMPC, and CMPC+
- TN3270 Server



Module VC/VR:
- Voice over IP
- Call Control Signalling
- Voice over Frame Relay
- Voice over ATM
- Telephony Applications
- Trunk Management
- Fax, Video, and Modem Support



Module QC/QR:
- Packet Classification
- Congestion Management
- Congestion Avoidance
- Policing and Shaping
- Signalling
- Link Efficiency Mechanisms



Module XC/XR:
- Cisco IOS Switching Paths
- NetFlow Switching
- Multiprotocol Label Switching
- Multilayer Switching
- Multicast Distributed Switching
- Virtual LANs
- LAN Emulation

47954

Master Indexes

Two master indexes provide indexing information for the Cisco IOS software documentation set: an index for the configuration guides and an index for the command references. Individual books also contain a book-specific index.

The master indexes provide a quick way for you to find a command when you know the command name but not which module contains the command. When you use the online master indexes, you can click the page number for an index entry and go to that page in the online document.

Supporting Documents and Resources

The following documents and resources support the Cisco IOS software documentation set:

- *Cisco IOS Command Summary* (two volumes)—This publication explains the function and syntax of the Cisco IOS software commands. For more information about defaults and usage guidelines, refer to the Cisco IOS command reference publications.
- *Cisco IOS System Error Messages*—This publication lists and describes Cisco IOS system error messages. Not all system error messages indicate problems with your system. Some are purely informational, and others may help diagnose problems with communications lines, internal hardware, or the system software.
- *Cisco IOS Debug Command Reference*—This publication contains an alphabetical listing of the **debug** commands and their descriptions. Documentation for each command includes a brief description of its use, command syntax, usage guidelines, and sample output.
- *Dictionary of Internetworking Terms and Acronyms*—This Cisco publication compiles and defines the terms and acronyms used in the internetworking industry.
- New feature documentation—The Cisco IOS software documentation set documents the mainline release of Cisco IOS software (for example, Cisco IOS Release 12.2). New software features are introduced in early deployment releases (for example, the Cisco IOS "T" release train for 12.2, 12.2(x)T). Documentation for these new features can be found in standalone documents called "feature modules." Feature module documentation describes new Cisco IOS software and hardware networking functionality and is available on Cisco.com and the Documentation CD-ROM.
- Release notes—This documentation describes system requirements, provides information about new and changed features, and includes other useful information about specific software releases. See the section "Using Software Release Notes" in the chapter "Using Cisco IOS Software" for more information.
- Caveats documentation—This documentation provides information about Cisco IOS software defects in specific software releases.
- RFCs—RFCs are standards documents maintained by the Internet Engineering Task Force (IETF). Cisco IOS software documentation references supported RFCs when applicable. The full text of referenced RFCs may be obtained on the World Wide Web at http://www.rfc-editor.org/.
- MIBs—MIBs are used for network monitoring. For lists of supported MIBs by platform and release, and to download MIB files, see the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

New and Changed Information

The *Cisco IOS Voice, Video, and Fax Configuration Guide* is the result of reorganizing and renaming the *Cisco IOS Multiservice Applications Configuration Guide*.

The *Cisco IOS Voice, Video, and Fax Configuration Guide* contains information about configuring Voice over IP, Voice over Frame Relay, Voice over ATM, and telephony applications using Interactive Voice Response (IVR), fax, and video. This release adds the following new technologies:

- Media Gateway Control Protocol/Simple Gateway Control Protocol
- Session Initiation Protocol
- T.38-compliant fax relay
- Hoot and holler over IP
- Caller ID

This release of the *Cisco IOS Voice, Video, and Fax Configuration Guide* deletes the following technologies:

- Broadband—covered in a separate configuration guide.
- Voice over HDLC—no longer supported by Cisco routers.

Document Conventions

Within Cisco IOS software documentation, the term *router* is generally used to refer to a variety of Cisco products (for example, routers, access servers, and switches). Routers, access servers, and other networking devices that support Cisco IOS software are shown interchangeably within examples. These products are used only for illustrative purposes; that is, an example that shows one product does not necessarily indicate that other products are not supported.

The Cisco IOS documentation set uses the following conventions:

Convention	Description
^ or Ctrl	The ^ and Ctrl symbols represent the Control key. For example, the key combination ^D or Ctrl-D means hold down the Control key while you press the D key. Keys are indicated in capital letters but are not case sensitive.
string	A string is a nonquoted set of characters shown in italics. For example, when setting an SNMP community string to public, do not use quotation marks around the string or the string will include the quotation marks.

Command syntax descriptions use the following conventions:

Convention	Description
boldface	Boldface text indicates commands and keywords that you enter literally as shown.
italics	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional element (keyword or argument).
|	A vertical line indicates a choice within an optional or required set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical line indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical line indicate a required choice.

Nested sets of square brackets or braces indicate optional or required choices within optional or required elements. For example:

Convention	Description
[x {y | z}]	Braces and a vertical line within square brackets indicate a required choice within an optional element.

Examples use the following conventions:

Convention	Description
`screen`	Examples of information displayed on the screen are set in Courier font.
`boldface screen`	Examples of text that you must enter are set in Courier bold font.
< >	Angle brackets enclose text that is not printed to the screen, such as passwords.
!	An exclamation point at the beginning of a line indicates a comment line. (Exclamation points are also displayed by the Cisco IOS software for certain processes.)
[ ]	Square brackets enclose default responses to system prompts.

The following conventions are used to attract the attention of the reader:

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Note Means *reader take note*. Notes contain helpful suggestions or references to materials not contained in this manual.

Timesaver Means the *described action saves time*. You can save time by performing the action described in the paragraph.

Cisco Connection Online

Cisco Connection Online (CCO) is Cisco Systems' primary, real-time support channel. Maintenance customers and partners can self-register on CCO to obtain additional information and services.

Available 24 hours a day, 7 days a week, CCO provides a wealth of standard and value-added services to Cisco's customers and business partners. CCO services include product information, product documentation, software updates, release notes, technical tips, the Bug Navigator, configuration notes, brochures, descriptions of service offerings, and download access to public and authorized files.

CCO serves a wide variety of users through two interfaces that are updated and enhanced simultaneously: a character-based version and a multimedia version that resides on the World Wide Web (WWW). The character-based CCO supports Zmodem, Kermit, Xmodem, FTP, and Internet e-mail, and it is excellent for quick access to information over lower bandwidths. The WWW version of CCO provides richly formatted documents with photographs, figures, graphics, and video, as well as hyperlinks to related information.

You can access CCO in the following ways:

- WWW: http://www.cisco.com
- WWW: http://www-europe.cisco.com
- WWW: http://www-china.cisco.com
- Telnet: cco.cisco.com
- Modem: From North America, 408 526-8070; from Europe, 33 1 64 46 40 82. Use the following terminal settings: VT100 emulation; databits: 8; parity: none; stop bits: 1; and connection rates up to 28.8 kbps.

For a copy of CCO's Frequently Asked Questions (FAQ), contact cco-help@cisco.com. For additional information, contact cco-team@cisco.com.

Note If you are a network administrator and need personal technical assistance with a Cisco product that is under warranty or covered by a maintenance contract, contact Cisco's Technical Assistance Center (TAC) at 800 553-2447, 408 526-7209, or tac@cisco.com. To obtain general information about Cisco Systems, Cisco products, or upgrades, contact 800 553-6387, 408 526-7208, or cs-rep@cisco.com.

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM, a member of the Cisco Connection Family, is updated monthly. Therefore, it might be more current than printed documentation. To order additional copies of the Documentation CD-ROM, contact your local sales representative or call customer service. The CD-ROM package is available as a single package or as an annual subscription. You can also access Cisco documentation on the World Wide Web at http://www.cisco.com, http://www-china.cisco.com, or http://www-europe.cisco.com.

If you are reading Cisco product documentation on the World Wide Web, you can submit comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco. We appreciate your comments.

Obtaining Documentation

The following sections provide sources for obtaining documentation from Cisco Systems.

World Wide Web

The most current Cisco documentation is available on the World Wide Web at the following website:

http://www.cisco.com

Translated documentation is available at the following website:

http://www.cisco.com/public/countries_languages.html

Ordering Documentation

Cisco documentation can be ordered in the following ways:

- Registered Cisco Direct Customers can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl

- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription

- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco corporate headquarters (California, USA) at 408 526-7208 or, in North America, by calling 800 553-NETS(6387).

Documentation Feedback

If you are reading Cisco product documentation on the World Wide Web, you can submit technical comments electronically. Click **Feedback** in the toolbar and select **Documentation**. After you complete the form, click **Submit** to send it to Cisco.

You can e-mail your comments to bug-doc@cisco.com.

To submit your comments by mail, use the response card behind the front cover of your document, or write to the following address:

Cisco Systems, Inc.
Document Resource Connection
170 West Tasman Drive
San Jose, CA 95134-9883

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain documentation, troubleshooting tips, and sample configurations from online tools. For Cisco.com registered users, additional troubleshooting tools are available from the TAC website.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information and resources at anytime, from anywhere in the world. This highly integrated Internet application is a powerful, easy-to-use tool for doing business with Cisco.

Cisco.com provides a broad range of features and services to help customers and partners streamline business processes and improve productivity. Through Cisco.com, you can find information about Cisco and our networking solutions, services, and programs. In addition, you can resolve technical issues with online technical support, download and test software packages, and order Cisco learning materials and merchandise. Valuable online skill assessment, training, and certification programs are also available.

Customers and partners can self-register on Cisco.com to obtain additional personalized information and services. Registered users can order products, check on the status of an order, access technical support, and view benefits specific to their relationships with Cisco.

To access Cisco.com, go to the following website:

http://www.cisco.com

Technical Assistance Center

The Cisco TAC website is available to all customers who need technical assistance with a Cisco product or technology that is under warranty or covered by a maintenance contract.

Contacting TAC by Using the Cisco TAC Website

If you have a priority level 3 (P3) or priority level 4 (P4) problem, contact TAC by going to the TAC website:

http://www.cisco.com/tac

P3 and P4 level problems are defined as follows:

- P3—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- P4—You need information or assistance on Cisco product capabilities, product installation, or basic product configuration.

In each of the above cases, use the Cisco TAC website to quickly find answers to your questions.

To register for Cisco.com, go to the following website:

http://www.cisco.com/register/

If you cannot resolve your technical issue by using the TAC online resources, Cisco.com registered users can open a case online by using the TAC Case Open tool at the following website:

http://www.cisco.com/tac/caseopen

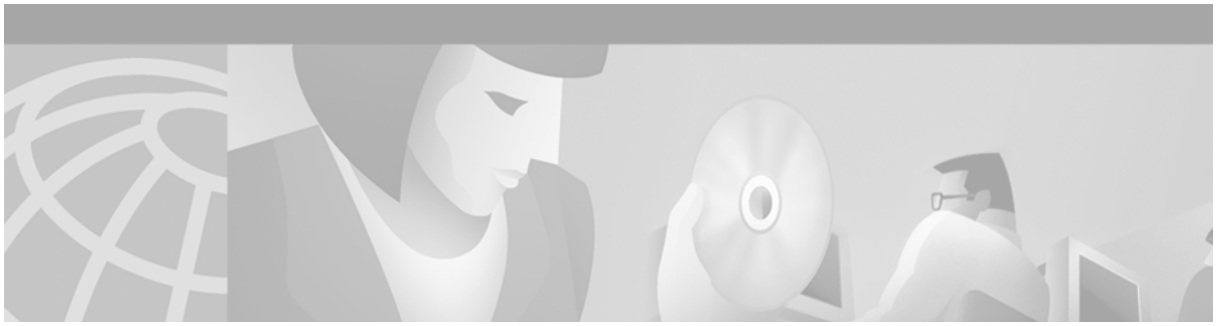
Contacting TAC by Telephone

If you have a priority level 1 (P1) or priority level 2 (P2) problem, contact TAC by telephone and immediately open a case. To obtain a directory of toll-free numbers for your country, go to the following website:

http://www.cisco.com/warp/public/687/Directory/DirTAC.shtml

P1 and P2 level problems are defined as follows:

- P1—Your production network is down, causing a critical impact to business operations if service is not restored quickly. No workaround is available.
- P2—Your production network is severely degraded, affecting significant aspects of your business operations. No workaround is available.



Using Cisco IOS Software

This chapter provides helpful tips for understanding and configuring Cisco IOS software using the command-line interface (CLI). It contains the following sections:

For an overview of Cisco IOS software configuration, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information on the conventions used in the Cisco IOS software documentation set, see the chapter "About Cisco IOS Software Documentation" located at the beginning of this book.

Understanding Command Modes

You use the CLI to access Cisco IOS software. Because the CLI is divided into many different modes, the commands available to you at any given time depend on the mode you are currently in. Entering a question mark (**?**) at the CLI prompt allows you to obtain a list of commands available for each command mode.

When you log in to the CLI, you are in user EXEC mode. User EXEC mode contains only a limited subset of commands. To have access to all commands, you must enter privileged EXEC mode, normally by using a password. From privileged EXEC mode you can issue any EXEC command—user or privileged mode—or you can enter global configuration mode. Most EXEC commands are one-time commands. For example, **show** commands show important status information, and **clear** commands clear counters or interfaces. The EXEC commands are not saved when the software reboots.

Configuration modes allow you to make changes to the running configuration. If you later save the running configuration to the startup configuration, these changed commands are stored when the software is rebooted. To enter specific configuration modes, you must start at global configuration mode. From global configuration mode, you can enter interface configuration mode and a variety of other modes, such as protocol-specific modes.

ROM monitor mode is a separate mode used when the Cisco IOS software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode.

Table 1 describes how to access and exit various common command modes of the Cisco IOS software. It also shows examples of the prompts displayed for each mode.

Table 1 Accessing and Exiting Command Modes

Command Mode	Access Method	Prompt	Exit Method
User EXEC	Log in.	`Router>`	Use the **logout** command.
Privileged EXEC	From user EXEC mode, use the **enable** EXEC command.	`Router#`	To return to user EXEC mode, use the **disable** command.
Global configuration	From privileged EXEC mode, use the **configure terminal** privileged EXEC command.	`Router(config)#`	To return to privileged EXEC mode from global configuration mode, use the **exit** or **end** command, or press **Ctrl-Z**.
Interface configuration	From global configuration mode, specify an interface using an **interface** command.	`Router(config-if)#`	To return to global configuration mode, use the **exit** command. To return to privileged EXEC mode, use the **end** command, or press **Ctrl-Z**.
ROM monitor	From privileged EXEC mode, use the **reload** EXEC command. Press the **Break** key during the first 60 seconds while the system is booting.	`>`	To exit ROM monitor mode, use the **continue** command.

For more information on command modes, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Getting Help

Entering a question mark (**?**) at the CLI prompt displays a list of commands available for each command mode. You can also get a list of keywords and arguments associated with any command by using the context-sensitive help feature.

To get help specific to a command mode, a command, a keyword, or an argument, use one of the following commands:

Command	Purpose
help	Provides a brief description of the help system in any command mode.
abbreviated-command-entry`?`	Provides a list of commands that begin with a particular character string. (No space between command and question mark.)
abbreviated-command-entry<**Tab**>	Completes a partial command name.
?	Lists all commands available for a particular command mode.
command **?**	Lists the keywords or arguments that you must enter next on the command line. (Space between command and question mark.)

Example: How to Find Command Options

This section provides an example of how to display syntax for a command. The syntax can consist of optional or required keywords and arguments. To display keywords and arguments for a command, enter a question mark (**?**) at the configuration prompt or after entering part of a command followed by a space. The Cisco IOS software displays a list and brief description of available keywords and arguments. For example, if you were in global configuration mode and wanted to see all the keywords or arguments for the **arap** command, you would type **arap ?**.

The <cr> symbol in command help output stands for "carriage return." On older keyboards, the carriage return key is the Return key. On most modern keyboards, the carriage return key is the Enter key. The <cr> symbol at the end of command help output indicates that you have the option to press **Enter** to complete the command and that the arguments and keywords in the list preceding the <cr> symbol are optional. The <cr> symbol by itself indicates that no more arguments or keywords are available and that you must press **Enter** to complete the command.

Table 2 shows examples of how you can use the question mark (**?**) to assist you in entering commands. The table steps you through configuring an IP address on a serial interface on a Cisco 7206 router that is running Cisco IOS Release 12.0(3).

Table 2 How to Find Command Options

Command	Comment

`Router> enable`
`Password: <password>`
`Router#`
	Enter the **enable** command and password to access privileged EXEC commands. You are in privileged EXEC mode when the prompt changes to `Router#`.

`Router# configure terminal`
`Enter configuration commands, one per line. End with CNTL/Z.`
`Router(config)#`
	Enter the **configure terminal** privileged EXEC command to enter global configuration mode. You are in global configuration mode when the prompt changes to `Router(config)#`.

`Router(config)# interface serial ?`
`  <0-6>  Serial interface number`
`Router(config)# interface serial 4 ?`
`  /`
`Router(config)# interface serial 4/ ?`
`  <0-3>  Serial interface number`
`Router(config)# interface serial 4/0`
`Router(config-if)#`
	Enter interface configuration mode by specifying the serial interface that you want to configure using the **interface serial** global configuration command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter the serial interface slot number and port number, separated by a forward slash.
	You are in interface configuration mode when the prompt changes to `Router(config-if)#`.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ?`
`Interface configuration commands:`
`  .`
`  .`
`  .`
`  ip  Interface Internet Protocol config commands`
`  keepalive  Enable keepalive`
`  lan-name  LAN Name command`
`  llc2  LLC2 Interface Subcommands`
`  load-interval  Specify interval for load calculation for an interface`
`  locaddr-priority  Assign a priority group`
`  logging  Configure logging for interface`
`  loopback  Configure internal loopback on an interface`
`  mac-address  Manually set interface MAC address`
`  mls  mls router sub/interface commands`
`  mpoa  MPOA interface configuration commands`
`  mtu  Set the interface Maximum Transmission Unit (MTU)`
`  netbios  Use a defined NETBIOS access list or enable name-caching`
`  no  Negate a command or set its defaults`
`  nrzi-encoding  Enable use of NRZI encoding`
`  ntp  Configure NTP`
`  .`
`  .`
`  .`
`Router(config-if)#`
	Enter **?** to display a list of all the interface configuration commands available for the serial interface. This example shows only some of the available interface configuration commands.

`Router(config-if)# ip ?`
`Interface IP configuration subcommands:`
`  access-group  Specify access control for packets`
`  accounting  Enable IP accounting on this interface`
`  address  Set the IP address of an interface`
`  authentication  authentication subcommands`
`  bandwidth-percent  Set EIGRP bandwidth limit`
`  broadcast-address  Set the broadcast address of an interface`
`  cgmp  Enable/disable CGMP`
`  directed-broadcast  Enable forwarding of directed broadcasts`
`  dvmrp  DVMRP interface commands`
`  hello-interval  Configures IP-EIGRP hello interval`
`  helper-address  Specify a destination address for UDP broadcasts`
`  hold-time  Configures IP-EIGRP hold time`
`  .`
`  .`
`  .`
`Router(config-if)# ip`
	Enter the command that you want to configure for the interface. This example uses the **ip** command.
	Enter **?** to display what you must enter next on the command line. This example shows only some of the available interface IP configuration commands.

Table 2 How to Find Command Options (continued)

Command	Comment

`Router(config-if)# ip address ?`
`  A.B.C.D  IP address`
`  negotiated  IP Address negotiated over PPP`
`Router(config-if)# ip address`
	Enter the command that you want to configure for the interface. This example uses the **ip address** command.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP address or the **negotiated** keyword.
	A carriage return (<cr>) is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 ?`
`  A.B.C.D  IP subnet mask`
`Router(config-if)# ip address 172.16.0.1`
	Enter the keyword or argument you want to use. This example uses the 172.16.0.1 IP address.
	Enter **?** to display what you must enter next on the command line. In this example, you must enter an IP subnet mask.
	A <cr> is not displayed; therefore, you must enter additional keywords or arguments to complete the command.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0 ?`
`  secondary  Make this IP address a secondary address`
`  <cr>`
`Router(config-if)# ip address 172.16.0.1 255.255.255.0`
	Enter the IP subnet mask. This example uses the 255.255.255.0 IP subnet mask.
	Enter **?** to display what you must enter next on the command line. In this example, you can enter the **secondary** keyword, or you can press **Enter**.
	A <cr> is displayed; you can press **Enter** to complete the command, or you can enter another keyword.

`Router(config-if)# ip address 172.16.0.1 255.255.255.0` `Router(config-if)#`	In this example, Enter is pressed to complete the command.

Using the no and default Forms of Commands

Almost every configuration command has a **no** form. In general, use the **no** form to disable a function. Use the command without the **no** keyword to reenable a disabled function or to enable a function that is disabled by default. For example, IP routing is enabled by default. To disable IP routing, use the **no ip routing** command; to reenable IP routing, use the **ip routing** command. The Cisco IOS software command reference publications provide the complete syntax for the configuration commands and describe what the **no** form of a command does.

Configuration commands also can have a **default** form, which returns the command settings to the default values. Most commands are disabled by default, so in such cases using the **default** form has the same result as using the **no** form of the command. However, some commands are enabled by default and have variables set to certain default values. In these cases, the **default** form of the command enables the command and sets the variables to their default values. The Cisco IOS software command reference publications describe the effect of the **default** form of a command if the command functions differently than the **no** form.

Saving Configuration Changes

Use the **copy system:running-config nvram:startup-config** command to save your configuration changes to the startup configuration so that the changes will not be lost if the software reloads or a power outage occurs. For example:

```
Router# copy system:running-config nvram:startup-config
Building configuration...
```

It might take a minute or two to save the configuration. After the configuration has been saved, the following output appears:

```
[OK]
Router#
```

On most platforms, this task saves the configuration to NVRAM. On the Class A Flash file system platforms, this task saves the configuration to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Filtering Output from the show and more Commands

In Cisco IOS Release 12.0(1)T and later releases, you can search and filter the output of **show** and **more** commands. This functionality is useful if you need to sort through large amounts of output or if you want to exclude output that you need not see.

To use this functionality, enter a **show** or **more** command followed by the "pipe" character (|); one of the keywords **begin**, **include**, or **exclude**; and a regular expression on which you want to search or filter (the expression is case-sensitive):

command **|** {**begin | include | exclude**} *regular-expression*

The output matches certain lines of information in the configuration file. The following example illustrates how to use output modifiers with the **show interface** command when you want the output to include only lines in which the expression "protocol" appears:

```
Router# show interface | include protocol

FastEthernet0/0 is up, line protocol is up
Serial4/0 is up, line protocol is up
Serial4/1 is up, line protocol is up
Serial4/2 is administratively down, line protocol is down
Serial4/3 is administratively down, line protocol is down
```

For more information on the search and filter functionality, refer to the "Using the Command-Line Interface" chapter in the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Identifying Supported Platforms

Cisco IOS software is packaged in feature sets consisting of software images that support specific platforms. The feature sets available for a specific platform depend on which Cisco IOS software images are included in a release. To identify the set of software images available in a specific release or to find out if a feature is available in a given Cisco IOS software image, see the following sections:

- Using Feature Navigator
- Using Software Release Notes

Using Feature Navigator

Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a particular set of features and which features are supported in a particular Cisco IOS image.

Feature Navigator is available 24 hours a day, 7 days a week. To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, e-mail the Contact Database Administration group at cdbadmin@cisco.com. If you do not have an account on Cisco.com, go to http://www.cisco.com/register and follow the directions to establish an account.

To use Feature Navigator, you must have a JavaScript-enabled web browser such as Netscape 3.0 or later, or Internet Explorer 4.0 or later. Internet Explorer 4.0 always has JavaScript enabled. To enable JavaScript for Netscape 3.x or Netscape 4.x, follow the instructions provided with the web browser. For JavaScript support and enabling instructions for other browsers, check with the browser vendor.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. You can access Feature Navigator at the following URL:

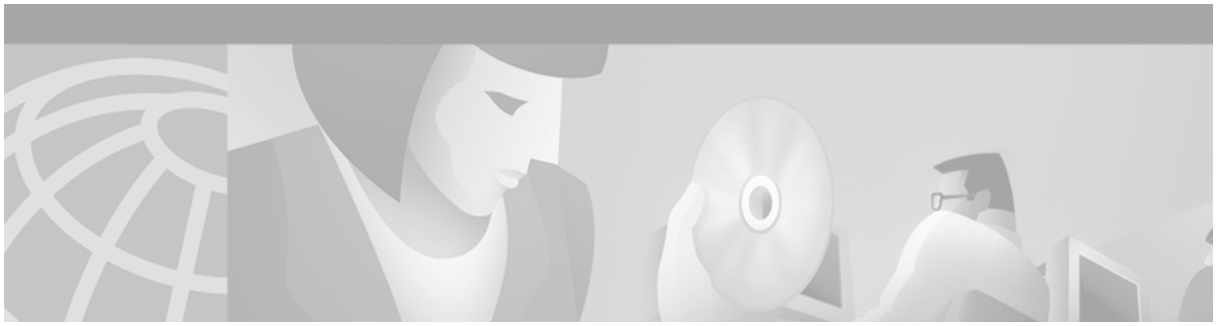
http://www.cisco.com/go/fn

Using Software Release Notes

Cisco IOS software releases include release notes that provide the following information:

- Platform support information
- Memory recommendations
- Microcode support information
- Feature set tables
- Feature descriptions
- Open and resolved severity 1 and 2 caveats for all platforms

Release notes are intended to be release-specific for the most current release, and the information provided in these documents may not be cumulative in providing information about features that first appeared in previous releases.



Voice, Video, and Fax Overview

The *Cisco IOS Voice, Video, and Fax Configuration Guide* shows you how to configure your Cisco router or access server to support voice, video, and fax applications. This chapter is an overview of some of the concepts and technologies described in the guide.

Configuration Guide Overview

The *Cisco IOS Voice, Video, and Fax Configuration Guide* is the result of reorganizing and renaming the *Cisco IOS Multiservice Applications Configuration Guide*. The reorganized publication is divided into the following parts:

- Basic Voice Configuration
- H.323 Support and Other VoIP Call Control Signaling Protocols
- Voice over Layer 2 Protocols
- Telephony Applications
- Trunk Management and Conditioning Features
- Fax, Video, and Modem Support

Each part contains one or more chapters that describe configuration procedures for each respective technology. The following sections describe some of the chapter contents for this configuration guide.

Dial Peers

Dial peers describe the entities to or from which a call is established and the key to understanding the Cisco voice implementation. All voice technologies use dial peers to define the characteristics associated with a call leg. A call leg is a discrete segment of a call connection that lies between two points in the connection. An end-to-end call comprises four call legs, two from the perspective of the source route, and two from the perspective of the destination route.

You use dial peers to apply specific attributes to call legs and to identify call origin and destination. Attributes applied to a call leg include specific quality of service (QoS) features (such as IP RTP Priority and IP Precedence), compression/decompression (codec), voice activity detection (VAD), and fax rate.

There are basically two different kinds of dial peers with each voice implementation:

- Plain old telephone service (POTS)—Dial peer describing the characteristics of a traditional telephony network connection. POTS peers point to a particular voice port on a voice network device.

  When you configure POTS dial peers, the key commands that you must be configure are the **port** and **destination-pattern** commands. The **destination-pattern** command defines the telephone number associated with the POTS dial peer. The **port** command associates the POTS dial peer with a specific logical dial interface, normally the voice port connecting the Cisco device to the local POTS network.

  Specific applications, such as interactive voice response (IVR), are configured on the POTS dial peer as well.

- Voice network (VoIP, VoATM, and VoFR)—Dial peer describing the characteristics of a packet network connection; in the case of VoIP, for example, it is an IP network. Voice-network peers point to specific voice-network devices.

  When you configure voice-network dial peers, the key commands that you must configure are the **destination-pattern** and **session-target** commands. The **destination-pattern** command defines the telephone number associated with the voice-network dial peer. The **session-target** command specifies a destination address for the voice-network peer.

Other applications (such as store-and-forward fax, which uses the infrastructure of VoIP but is not strictly a voice technology) also use dial peers to assign attributes to call legs.

Voice Ports

Voice port commands define the characteristics associated with a particular voice-port signaling type. The Cisco implementation of voice supports both analog and digital telephony connections. The connection supported (and the associated signaling) depends on the type of voice network module (VNM) or voice feature card (VFC) installed in your Cisco router or access server.

Voice ports provide support for three basic analog voice signaling formats:

- FXO—Foreign Exchange Office interface. The FXO interface is an RJ-11 connector that allows a connection to be directed at the Public Switched Telephone Network (PSTN) central office (CO) (or to a standard PBX interface, if the local telecommunications authority permits). This interface is of value for off-premises extension applications.
- FXS—Foreign Exchange Station interface. The FXS interface is an RJ-11 connector that allows connection for basic telephone equipment, keysets, and PBXs; FXS connections supply ring, voltage, and dial tone.
- E&M—Ear and mouth (or recEive and transMit) interface. The E&M interface is an RJ-48 connector that allows connection for PBX trunk lines (tie lines). It is a signaling technique for 2-wire and 4-wire telephone and trunk interfaces.

  The Cisco MC3810 multiservice concentrator also supports E&M Mercury Exchange Limited channel-associated signaling (MEL CAS), which is used primarily in the United Kingdom.

Depending on the Cisco device you are configuring, the following digital signaling is supported:

- ISDN PRI
- ISDN BRI

- E1 R2
- T1 CAS

The voice port syntax depends on the hardware platform on which it is being configured.

Voice Technologies

Cisco IOS Release 12.2 offers the following voice and voice-related technologies:

- VoIP
- Voice over Frame Relay (VoFR)
- Voice over ATM (VoATM)
- H.323 gateways
- Media Gateway Control Protocol (MGCP) and related protocols
- Session Initiation Protocol (SIP)
- Tool Command Language (TCL) and interactive voice response (IVR)
- Multimedia Conference Manager
- Fax gateways
- Video

Voice over IP

Cisco offers VoIP that uses IP to carry voice traffic. Because voice traffic is being transported via IP, you need to configure signaling parameters as part of the voice-port configuration in addition to feature-specific elements such as dial peers. VoIP is compliant with International Telecommunications Union-Telecommunications (ITU-T) specifications H.323 and Cisco's H.323 Version 2.

VoIP can be used to provide the following:

- A central-site telephony termination facility for VoIP traffic from multiple voice-equipped remote office facilities.
- A PSTN gateway for Internet telephone traffic. VoIP used as a PSTN gateway leverages the standardized use of H.323-based Internet telephone client applications. In the case of a device with extensive capacity running VoIP (such as the Cisco AS5800 universal access server), it provides the functionality of a carrier class switch.

VoIP enables Cisco routers and access servers to carry voice traffic (for example, telephone calls and faxes) over an IP network. In VoIP, the digital signal processor (DSP) segments the voice signal into frames that are then coupled in groups of two and stored in voice packets. The voice packets are transported using IP in compliance with ITU-T specification H.323. Because VoIP is a delay-sensitive application, you must have a well-engineered network end-to-end to use VoIP successfully. Fine-tuning your network to adequately support VoIP involves a series of protocols and features geared toward QoS. Traffic shaping considerations must be taken into account to ensure the reliability of the voice connection.

Voice over Frame Relay

VoFR uses Frame Relay to transport voice traffic. Because VoFR is transporting signals over Layer 2, you must configure timing parameters in addition to feature-specific elements such as dial peers and voice ports. VoFR is compliant with FRF.11 and FRF.12 specifications.

VoFR enables a Cisco device to carry voice traffic (for example, telephone calls and faxes) over a Frame Relay network. When voice traffic is sent over Frame Relay, the voice traffic is segmented and encapsulated for transit across the Frame Relay network. The segmentation engine uses FRF.12 fragmentation. FRF.12 (also known as FRF.11 Annex C) allows long data frames to be fragmented into smaller pieces and interleaved with real-time frames. In this way, real-time voice and nonreal-time data frames can be carried together on lower speed links without causing excessive delay to the real-time traffic.

The segmentation size configured must match the line rate or port access rate. To ensure a stable voice connection, you must configure the same data segmentation size on both sides of the voice connection. When voice segmentation is configured, all priority queueing, custom queueing, and weighted fair queueing are disabled on the interface.

When you configure voice and data traffic over the same Frame Relay data-link connection identifier (DLCI), you must take traffic-shaping considerations into account to ensure the reliability of the voice connection.

Cisco VoFR implementation supports the following types of VoFR calls:

- Static FRF.11 trunks
- Switched VoFR calls:
  - Dynamic switched calls
  - Cisco trunk (private line) calls

Voice over ATM

VoATM uses ATM adaptation layer 5 (AAL5) to route voice traffic. Because VoATM is transporting signals over Layer 2, you must configure timing parameters in addition to feature-specific elements such as dial peers and voice ports.

VoATM enables a Cisco MC3810 multiservice concentrator to carry voice traffic (for example, telephone calls and faxes) over an ATM network. The Cisco MC3810 multiservice concentrator supports compressed VoATM on ATM port 0 only.

When voice traffic is sent over ATM, the voice traffic is encapsulated using a special AAL5 encapsulation for multiplexed voice. The ATM permanent virtual circuit (PVC) must be configured to support real-time voice traffic, and the AAL5 voice encapsulation must be assigned to the PVC. The PVC must also be configured to support variable bit rate (VBR) for real-time networks for traffic shaping between voice and data PVCs.

Traffic shaping is necessary so that the carrier does not discard the incoming calls from the Cisco MC3810 multiservice concentrator. To configure voice and data traffic shaping, you must configure the peak, average, and burst options for voice traffic. Configure the burst value if the PVC will be carrying bursty traffic. The peak, average, and burst values are needed so the PVC can effectively handle the bandwidth for the expected number of voice calls.

H.323 Gateways

The H.323 standard provides for sending audio, video, and data conferencing data on an IP-based internetwork. The Cisco functionality enables gateway H.323 terminals to communicate with terminals running other protocols. Gateways provide protocol conversion between terminals running different types of protocols. Gatekeepers are optional nodes that manage other nodes in an H.323 network. Gateways communicate with gatekeepers using the registration, admission, and status (RAS) protocol. The gatekeeper maintains resource computing information, which it uses to select the appropriate gateway during the admission of a call.

Cisco software complies with the mandatory requirements and several of the optional features of the H.323 Version 2 specification. Cisco H.323 Version 2 software enables gatekeepers, gateways, and proxies to send and receive all the required fields in H.323 Version 2 messages. Cisco H.323 Version 2 features include the following:

- Lightweight registration
- Improved gateway selection process
- Gateway resource availability reporting
- Support for single proxy configurations
- Tunneling of redirecting number information element
- H.245 tunneling
- Hookflash relay
- H.235 security
- Codec negotiation
- H.245 empty capabilities set

Media Gateway Control Protocol

Media Gateway Control Protocol (MGCP) defines the call control relationship between VoIP gateways that translate audio signals to and from the packet network and call agents (CAs). The CAs are responsible for processing the calls. The MGCP gateways interact with a CA, also called a Media Gateway Controller (MGC) that performs signal and call processing on gateway calls. In the MGCP configurations supported by Cisco, the gateway can be a Cisco router, access server, or cable modem, and the CA is a third-party server.

Session Initiation Protocol

Session Initiation Protocol (SIP) is an alternative protocol developed by the Internet Engineering Task Force (IETF) for multimedia conferencing over IP. SIP features are compliant with IETF RFC 2543, *SIP: Session Initiation Protocol*, published in March 1999.

The Cisco SIP functionality enables Cisco access platforms to signal the setup of voice and multimedia calls over IP networks. The SIP feature also provides nonproprietary advantages in the following areas:

- Protocol extensibility
- System scalability
- Personal mobility services
- Interoperability with different vendors

SIP is an ASCII-based, application-layer control protocol that can be used to establish, maintain, and terminate calls between two or more endpoints.

Like other VoIP protocols, SIP is designed to address the functions of signaling and session management within a packet telephony network. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call.

SIP provides the following capabilities:

- Determining the location of the target endpoint—SIP supports address resolution, name mapping, and call redirection.
- Determining the media capabilities of the target endpoint—Through Session Description Protocol (SDP), SIP determines the lowest level of common services between the endpoints. Conferences are established using only the media capabilities that can be supported by all endpoints.
- Determining the availability of the target endpoint—If a call cannot be completed because the target endpoint is unavailable, SIP determines whether the called party is connected to a call already or did not answer in the allotted number of rings. SIP then returns a message indicating why the target endpoint was unavailable.
- Establishing a session between the originating and target endpoints—If the call can be completed, SIP establishes a session between the endpoints. SIP also supports midcall changes, such as the addition of another endpoint to the conference or the changing of a media characteristic or codec.
- Handling the transfer and termination of calls—SIP supports the transfer of calls from one endpoint to another. During a call transfer, SIP simply establishes a session between the transferee and a new endpoint (specified by the transferring party) and terminates the session between the transferee and the transferring party. At the end of a call, SIP terminates the sessions among all parties.

Interactive Voice Response

IVR consists of simple voice prompting and digit collection to gather caller information for authenticating the user and identifying the destination. IVR applications can be assigned to specific ports or invoked on the basis of Dialed Number Identification Service (DNIS). An IP public switched telephone network gateway can have several IVR applications to accommodate many different gateway services, and you can customize the IVR applications to present different interfaces to the various callers.

IVR systems provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words, or more commonly dual tone multifrequency (DTMF) signaling. For example, when a user makes a call with a debit card, an IVR application is used to prompt the caller to enter a specific type of information, such as an account number. After playing the voice prompt, the IVR application collects the predetermined number of touch tones and then places the call to the destination phone or system.

IVR uses TCL scripts to gather information and to process accounting and billing. For example, a TCL IVR script plays when a caller receives a voice-prompt instruction to enter a specific type of information, such as a personal identification number (PIN). After playing the voice prompt, the TCL IVR application collects the predetermined number of touch tones and sends the collected information to an external server for user authentication and authorization.

Since the introduction of the Cisco IVR technology, the software has undergone several enhancements. Cisco TCL IVR Version 2.0 is made up of separate components that are described individually in the sections that follow. The enhancements are as follows:

- MGCP scripting package implementation
- Real Time Streaming Protocol (RTSP) client implementation

- TCL IVR prompt playout and digit collection on IP call legs
- New TCL verbs to use RTSP and MGCP scripting features

The enhancements add scalability and enable the TCL IVR scripting functionality on VoIP legs. In addition, support for RTSP enables VoIP gateways to play messages from RTSP-compliant announcement servers. The addition of these enhancements also reduces the CPU load and saves memory on the gateway because no packetization is involved. Larger prompts can be played, and the use of an external audio server is allowed.

Multimedia Conference Manager

The Multimedia Conference Manager provides both gatekeeper and proxy capabilities, which are required for service provisioning and management of H.323 networks. With Multimedia Conference Manager you can configure your current internetwork to route bit-intensive data such as audio, telephony, video and audio telephony, and data conferencing using existing telephone and ISDN links, without degrading the current level of service in the network. In addition, you can implement H.323-compliant applications on existing networks in an incremental fashion without upgrades.

With Multimedia Conference Manager, you can provide the following services:

- Identification of H.323 traffic and application of appropriate policies
- Limiting of H.323 traffic on LANs and WANs
- User accounting for records based on service utilization
- Insertion of QoS for the H.323 traffic generated by applications such as VoIP, data conferencing, and video conferencing
- Implementation of security for H.323 communications

Video

Cisco 2600 series, 3600 series, and 7200 series routers and the Cisco MC3810 multiservice concentrator support the H.323 gatekeeper (sometimes referred to as Multimedia Conference Manager) with voice gateway image with Resource Reservation Protocol (RSVP) to ATM SVC mapping. This feature delivers H.323 gatekeeper, proxy, and voice gateway solutions with routing as a single Cisco IOS image. In addition, it enables H.323 RSVP reservations to be mapped to ATM non-real-time variable bit rate (nRTVBR) SVCs to guarantee quality of service (QoS) for video applications over ATM backbones.

Cisco supports video traffic within a data stream in three ways:

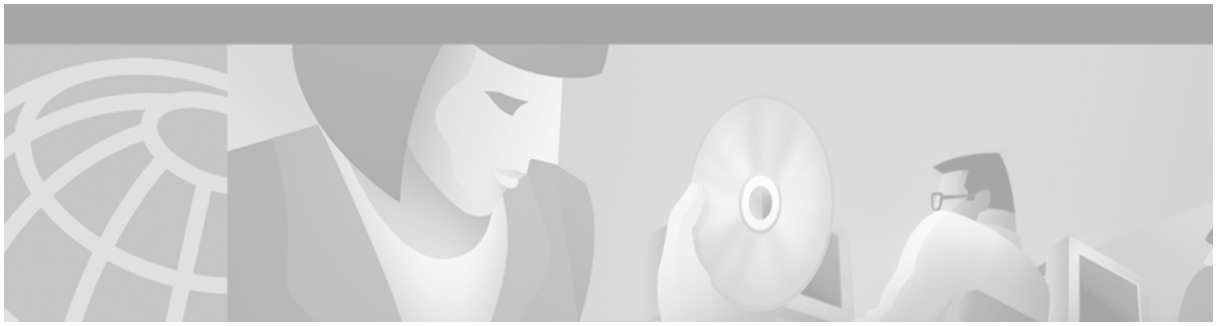
- Video in pass-through mode—Using this method, video traffic received from a video codec connected to a universal I/O serial port can be transported on a dedicated time slot between systems using the time-division multiplexing (TDM) functionality of the T1/E1 trunk.
- Video over ATM AAL1—A serial stream from a video codec connected to a serial port can be converted to ATM and transported across an ATM network using AAL1 circuit emulation service (CES) encapsulation.
- Video over ATM PVCs and switched virtual circuits (SVCs)—A serial stream from a video codec connected to a Cisco MC3810 multiservice concentrator using the plug-in Video Dialing Module (VDM) can be converted to ATM and transported across an ATM network using AAL1 CES.

Fax Gateways

Fax applications enable Cisco AS5300 universal access servers to send and receive faxes across packet-based networks using modems or VFCs. Some of the benefits of the fax gateway are as follows:

- Universal inbox for fax and e-mail—Faxes and e-mails can go to the same mailbox using DID numbers. E-mail and fax recipients can be combined.
- Toll bypass—In an enterprise environment in which offices in different cities are connected using a WAN, toll charges can be bypassed by transmitting faxes over the network connection. Because a fax message is stored on the mail server until Simple Mail Transfer Protocol (SMTP) forwards messages to the recipient, SMTP can forward fax e-mail attachments during off-peak hours (for example, during evenings and weekends), thereby reducing long-distance charges.
- Broadcast to multiple recipients—E-mail fax attachments can be sent to multiple recipients simultaneously.
- Improve robustness—The Fax Relay Packet Loss Concealment feature improves the robustness of the facsimile relay. It eliminates fax failures and lost data caused by excessive page errors. Field diagnostics and troubleshooting capabilities are improved by available debug commands. Statistics give better visibility into the real-time fax operation in the gateway, allowing for improved field diagnostics and troubleshooting.
- Cost savings and port density using T.37/T.38 Fax Gateway—The cost of maintaining one architecture (either fax or voice) is eliminated. Service providers can do the following:
  - Use a single port for voice, fax relay, and store-and-forward fax. For smaller points of presence (POPs), the single-port configuration for these technologies is even more significant because mixed traffic can be handled more efficiently, requiring only a single pool of ports versus splitting traffic across two pools.
  - Offer the new service of a single number for subscriber voice and fax access. The applications that use a single number for voice and fax require only half as many DNIS numbers and dial peers as would be required with separate voice and fax applications.
  - Offer applications that require toggling from voice to fax. Applications such as never-busy fax service can be addressed once the gateway can dynamically switch from fax relay to fax store and forward.
- Interoperability with T.37 fax relay for VoIP H.323—The Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrator gateways with ITU-T T.38 fax relay capability can interoperate with third-party gateways and gatekeepers over an IP H.323 network. The goal is to work with third-party gateways and gatekeepers to provide ITU-T standards-based T.38 fax relay services for multivendor networks.

The Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrator gateways provide standards-based toll bypass for fax and voice calls. In addition to existing voice and fax toll bypass capabilities, the multiservice gateways provide toll bypass for fax relay with the standards-based ITU-T T.38 fax relay implementation.



Cisco Voice Telephony

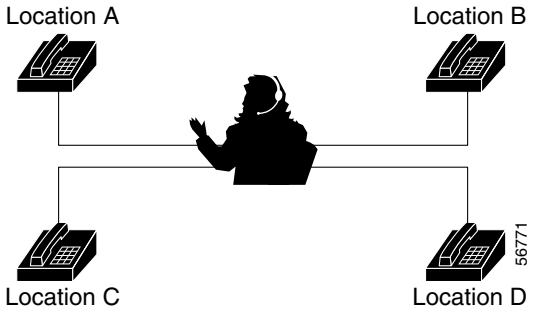
This chapter describes traditional telephony concepts and voice applications specific to Cisco IOS software. The chapter contains the following sections:

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Traditional Telephony

Telephony is the science of converting sound to electrical signals and transmitting those signals between points. Between the points are telephone offices with some kind of switching equipment. In the early days of the telephone, when the service was called plain old telephone service (POTS), the switching equipment was an operator (shown in Figure 2) who terminated the call on a console. The operator used patch cords to connect the two parties.

Figure 2 The Human Switch



The alternative to the human switch was automatic switching equipment that was invented by a man named Strowger and connected calls using trunks to and from the central office (CO). The earliest form of automatic switching was a rotary dial telephone in a subscriber's home, which was connected to the CO. Table 3 shows the history of switching technology.

Table 3 History of Switching Technology

Date	Switch Type	Operation	Switching Method	Control Type
1878	Manual	Manual	Space/analog	Human
1892	Step-by-step	Electromechanical	Space/analog	Distributed stage-by-stage
1918	Crossbar	Electromechanical	Space/analog	Common
1960	Electronic Switching System (ESS)	Semi-electronic	Space/analog	Common
1972	ESS (second generation)	Semi-electronic	Space/analog	Stored program
1976	ESS (third generation)	Electronic	Time/digital/PCM	Stored program

The CO could have a combination of Strowger switches, step-by-step switches, and crossbar switches, depending on the traffic volume. Automatic switching evolved into electronic, often called digital switching, which is widely used today.

POTS was also known as the loop-start service on the local loop. The Public Switched Telephone Network (PSTN) is the netire network over which POTS, ISDN, and other types of service are offered. The service evolved so that additional services could be offered to the average consumer, for example, call waiting, caller ID, and messaging. These services worked with any telephone.

Access to nationwide and international carriers was also provided from the CO using a complicated telephone network consisting of lines, trunks, and exchanges. The CO houses the switches and the switches make the connections for each call.

Prior to1980, AT&T owned most of the PSTN, and in theory was supposed to ensure universal service and low pricing for local calls. AT&T carried the traffic from the originating office through its extensive long distance network to the destination. In 1982, however, because of an anti-trust suit, AT&T was broken up into seven Regional Bell Operating Companies (RBOCs) and was prohibited from offering local telephone services.

Two types of carriers were created as a result of the anti-trust suit: Local Exchange Carriers (LECs) and IntereXchange Carriers (IXCs). The LECs made calls that originated and terminated in the same local area, while the IXCs transported calls that originated in one IXC and terminated outside the local area.

To define a local and an interexchange call, Local Access and Transport Areas (LATAs) mapped the geography of the US into local service areas. Calls that originate and terminate in the same LATA are carried by the LECs. If a call crosses a LATA boundary, it cannot be transported by the RBOCs, so the IXCs, such as AT&T, MCI, and Sprint, typically carry calls across the LATA boundaries.

The Telephone

There are three basic elements of voice communications: the telephone, switching equipment, and signaling. Switching equipment and signaling have been covered in previous sections. The telephone comes in many sizes, shapes, and colors, but all have these elements:

- Handset—The handset has a transmitter and receiver for converting sound waves into electrical signals that can be transmitted and received to and from COs and telephones.
- Switchhook—The term switchhook comes from the old-style telephones that had a hook on the side on which the handset was hung. Now the switchhook is a button in a cradle that the handset rests on when the telephone is idle.
- Keypad—The keypad has keys labeled with letters and numbers.

Anatomy of a Call

Not long ago, a call was placed using a local operator who would make the call. As telephone calling volumes grew, the process had to be automated. Rotary dials were added to the telephone that transferred the manual part of the dialing process to the user from the operator.

Rotary dials generate pulses on the telephone line by opening and closing an electrical circuit when the dial is turned and released. The number of pulses is determined by how far the dial is turned. When the dial is released, the pulses are generated as a spring rotates the dial back to the resting position. The pulses are generated at the rate of 10 pulses per second. Each pulse is 1/20 of a second long, with a 1/20 of a second pause between pulses. This is called out-pulsing.

Today, DTMF tone generators have been added in a keypad. The tones are generated when a button on the keypad is pressed. The electrical contacts are closed and cause two oscillators to generate two tones at specified frequencies. The combined tones are the signal for one of the digits. To be accepted by the CO, the dial tones must last at least 40 millseconds.

When the handset is lifted out of its cradle, the circuit is closed and electrical current flows to the CO. The CO responds with dial tone. This condition is called *off-hook*. When the handset rests in the cradle and on the switchhook, the circuit is open (no dial tone). This condition is called *on-hook*.

Table 4 shows the tones generated by the DTMF dial pad. Frequencies are combined to complete a call. For example, when the digit 6 is pressed, the tones belonging to frequencies 770 Hz and 1477 Hz are sent to the CO.

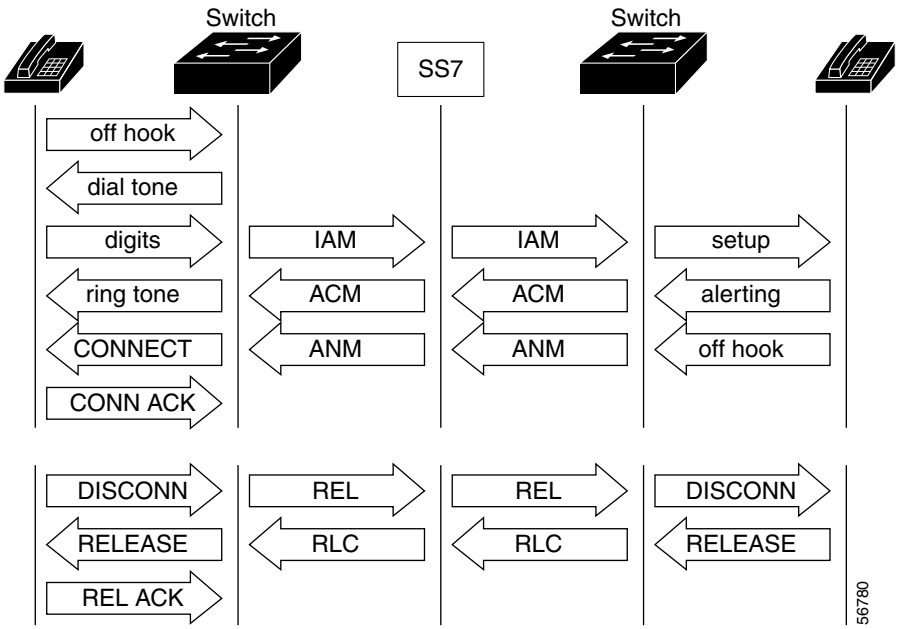
Table 4 DTMF (Touchtone) Pad

Hertz Level	**Corresponding Keypad Numbers**		
697 Hz	1	2	3
770 Hz	4	5	6
852 Hz	7	8	9
941 Hz	*	0	#
Hertz Level	1209 Hz	1336 Hz	1477 Hz

After the digits are received by the CO, they are stored in the switch. The first three digits of the called number determine whether the call is a local call or long distance. If it is local, the switch also determindes whether it can make the call by itself or if it has to forward to another CO. If it determines that the call is long distance, the CO forwards the call to a toll office, which in turn, forwards the call of another toll office or the local CO for completion. All of these steps take place in a matter of seconds.

The anatomy of a call is shown in Figure 3. The call traverses a CO switch, the SS7 network, and a second CO switch.

Figure 3 Anatomy of a Call



The following steps describe the call process shown in Figure 3:

Step 1 The handset goes off-hook and sends an off-hook indication to the end-office switch.

Step 2 The switch sends returns dial tone.

Step 3 The destination number is dialed (sent in-band using DTMF).

Step 4 The switch interprets the digits and sends an Initial Address Message (IAM), or setup message, to the SS7 network.

Step 5 The SS7 network reads the incoming IAM and sends a new IAM to the destination switch.

Step 6 The destination switch sends a setup message to the destination phone and it rings.

Step 7 When the phone rings, an alerting message is sent from the destination switch (not from the destination phone) back to the SS7 network through an Address Complete Message (ACM).

Step 8 The SS7 network reads the incoming ACM and generates an ACM to the originating switch.

Step 9 The calling party hears a ringing sound that means that the destination phone is ringing. The ringing is not synchronized; the local switch normally generates the ringing when the ACM is received from the SS7 network.)

Step 10 The called number goes off-hook (answers the incoming call) and sends an off-hook indication to its switch.

Step 11 The called number switch sends an ANswer Message (ANM) that is read by the SS7, and a new ANM is generated to originating switch.

Step 12 A connect message is sent, if the called number is an ISDN phone, and a connect acknowledgment is sent back, if the calling number is an ISDN phone. If either phone is not an ISDN phone, then on-hook or off-hook signal are sent to the end-office switch.

Step 13 A conversation is now possible between the two parties.

Voice Transmission Methods

There are four ways to transmit voice:

- Copper wire—Also known as unshielded twisted pair (UTP). The cable that was comprised of copper wire was not shielded, so part of signal leaked, primarily the high-frequency part of the signal. Even with the leakage, using copper wire was sufficient for voice transmissions. However, for high-speed data transmissions, UTP was inadequate because of the methods used to enhance the voice signal. The enhancements were:
  - Loading: Coils were added to loops longer than 18,000 feet. The signal was passed through the coils without attenuation and blocked frequencies above the voiceband.
  - Bridge taps: Bridge taps are unterminated portions of a loop not in the direct talking path. A bridge tap could be a cable pair connected at an intermediate point or an extension beyond the customer.
- Coaxial cable—Usually a single strand of copper running down the axis of the cable. The strand is separated from the outer shielding by an insulator made of foam or other non-conductive materials. A conducive shield covers the cable. Because of the construction of the cable, very high frequencies can be carried without leaking signal.

- Fiber optic—A strand or strands of glass that carry transmissions at a million times higher bandwidth than copper wire and coaxial cable. Fiber cable, however, is more expensive to install than copper wiring or coaxial cabling.
- Wireless—Several forms, such as microwave, synchronous satellites, low-earth-orbit satellites, cellular, and personal communications service (PCS). Wireless is usually categorized as fixed or modbile. Each form has obviated the need for a complex wired infrastructure.

Switching

Switching is known as the connection of the calling party to the called party and may involve one or many physical switches. As explained earlier, switches are mechanical, electrical, or electronic devices that open or close paths or circuits and are contained in offices. The offices are called switching offices, COs, end-offices, tandem offices, and toll offices. Offices are arranged in a hierarchy, as shown in Figure 4.

Higher-layer tandem offices connect local tandem offices. Many COs are directly connected to each other by trunks (lines or circuits). If enough traffic occurs between two COs, a dedicated circuit is placed between the two to offload those calls from the local-tandem offices. Some portions of the PSTN use as many as five levels of offices:

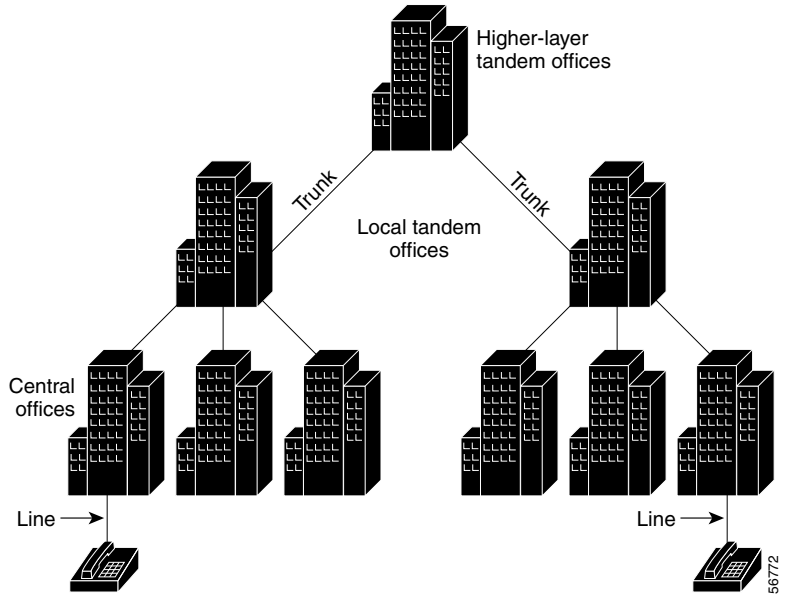
- Toll centers—1 through 4
- COs—5

A long-distance call begins at the subscriber's home and enters the network through the subscriber's CO. The call searches the office for an idle circuit. If the most direct route is busy, the call moves up the hierarchy to the next office, and the next, until a path is located to complete the call.

There are three switching occurrences that form the PSTN core:

- Local loops to COs—Physical cabling that connects the home telephone to the CO, also known as a Class 5 switch or end-office switch, using a telephone line.
- Tandem and toll offices—Networked offices configured in such a way as to open a communication path between several CO switches and the home phone. Trunks and lines connect the offices.
- International gateways—Offices that provide connectivity to networks around the world.

Figure 4 Circuit Switching Hierarchy



Switching Methods

The telephone network is a switched network consisting of a path or circuit connecting one telephone to another and maintained only for the duration of the call. Switching methods are as follows:

- Manual
- Step-by-step
- Crossbar
- Digital or ESS (first, second, and third generations)
- Packet
- Time-division multiplexing (TDM)

Digital and packet switching are described in the sections that follow.

Digital Switching

Digital switching is faster and less prone to hardware failure than analog switching. The most sophisticated COs are designed to have no more than one total failure in 40 years. The electronic CO can handle thousands of lines and larger service areas and require relatively little maintenance.

The typical digital switch has four essential components:

- Central processor—Controls call processing activities; for example, assigning time slots and administering features, such as call forwarding. Also directs system-control functions, system maintenance, and the loading and downloading of software.
- Switch matrix—Handles the actual connection of calls to their destinations. The latest switching modules can process up to 64,000 channels in a single cabinet and can switch wideband data as effortlessly as a voice conversation.

- Interfaces—Convert incoming voice and data signals into the digital format used by the switch and perform some low-level call processing tasks. Typical interfaces include those that terminate lines, trunks, digital loop carriers, and maintenance trunks.
- Input/output controllers—Provide access to the switch for maintenance, billing, routine operations and administration, and loading of software.

Fundamental call processing in a digital switch is a combination of the following elements:

- Call detection (off-hook detection)
- Dial tone transmission
- Digit collection and translation (including interpreting rotary-dialed digits and DTMF digits)
- Call routing
- Call connection
- Audible ringing and ringback
- Speech path establishment
- Call termination (on-hook detection)

Packet Switching

Packet switching is a networking switching method in which nodes share bandwidth with each other by sending packets of voice and data on the most efficient path and enabling a communications channel to be shared by multiple connections.

A packet-switched network requires a modified desktop, gateway, and gatekeeper. If multiparty conferencing is required, a multipoint controller unit (MCU) is also required. On the desktop, an IP phone that plugs into an RJ-45 jack or a handset or headset that plugs into a PC is required. The gateway is designed to convert voice from the packet to the circuit-switched domain. Signaling information, including dial tone, is passed by the gateway. Five types of connections are supported:

- Analog
- Digital ISDN T1, T3, or E1
- Digital ISDN (PRI or BRI)
- ATM at OC-3 and higher speeds
- Frame Relay

A gatekeeper is designed to throttle the origination of additional real-time connections over the network. The real-time applications register with the gatekeeper before attempting to bring up a session. The gatekeeper has the authority to reject a request or grant one at a diminished data rate. Th authorization process is critical in video connections, which can consume vast amounts of bandwidth for high-quality connections. The gatekeeper controls and manages calls and provides voice-switching intelligence.

Time-Division Multiplexing

TDM is a signaling technique in which information from multiple channels can be allocated bandwidth on a single wire according to preassigned time slots. Bandwidth is allocated to each channel regardless of whether the station has data to transmit.

Each channel of information is an 8-bit digital signal that is combined into a 24-channel, 125-microsecond frame. The multiplexing enables many channels of information to be simultaneously transmitted over the same pathway as pieces of the signal are woven together one after the other and assigned time slots on the pathway.

Digital switches use TDM to process a huge number of calls in a small amount of time. The switching matrix uses TDM to place the incoming traffic onto the proper outgoing time slots to lines and trunks. After switching, the digital signals are multiplexed back together and sent to the called party.

Bandwidth

Bandwidth is the difference between the highest and lowest frequencies available for signaling. The term can also refer to the rated throughput capacity of a given network medium or protocol. In analog communications, bandwidth is typically measured in Hertz (Hz)—cycles per second. In digital communications, bandwidth is typically measured in bits per second (bps).

Voice signals can be filtered such that only frequencies above 300 Hz and below 3,300 Hz will pass and the information carried by the signal will not be impaired. Using a limited frequency range means that a CO and the customer's station equipment can accommodate a wide variety of signaling loops that interconnect them.

Signaling

The human voice generates sound waves, and the telephone converts the sound waves into electrical signals, analogous to sound. However, analog signaling is not robust, either because of line noise or inefficient techniques used to reduce line noise.

Analog transmissions are boosted by amplifiers, because the signal diminishes, depending on the distance it has to travel from the CO. As the signal is boosted, the noise is also boosted, which often causes an unusable connection.

In digital networks, signals are transmitted over great distances and coded, regenerated, and decoded without degradation of quality. Repeaters amplify the signal and clean it to its original condition. Repeaters then determine the original sequence of the signal levels and send the clean signal to the next network destination.

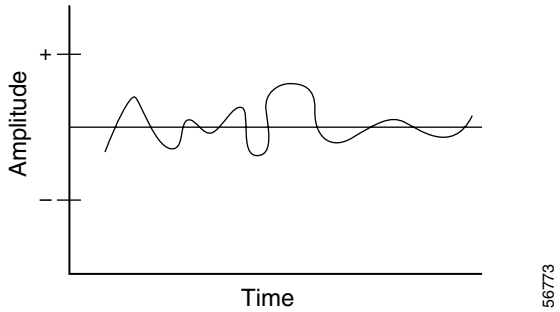
The following sections describe the following types of signaling:

- Analog Signaling, page 18
- Digital Signaling, page 18
- Call Control Signaling, page 19
- Line or Circuit Signaling, page 23
- Supervisory Signaling, page 24
- Q.931 Signaling, page 24
- QSIG Signaling, page 25

Analog Signaling

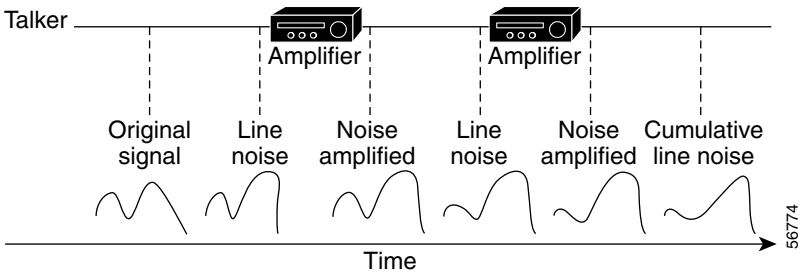
Analog signals consist of two elements, time and amplitude, and enable the transmission of voice-band information. Analog signaling is expressed in a waveform and can be seen on an oscilloscope. Figure 5 is a high-level view of an analog waveform.

Figure 5 Analog Waveform



Analog signals are susceptible to line noise that distort the analog waveform and cause garbled reception. This effect is especially more obvious to the listener if many amplifiers are located between the listener and the end-office switch. Figure 6 shows that an amplifier does not clean the signal as it amplifies, but simply amplifies the distorted signal. When the signal goes through several amplifiers, the noise that is created is called cumulative line noise.

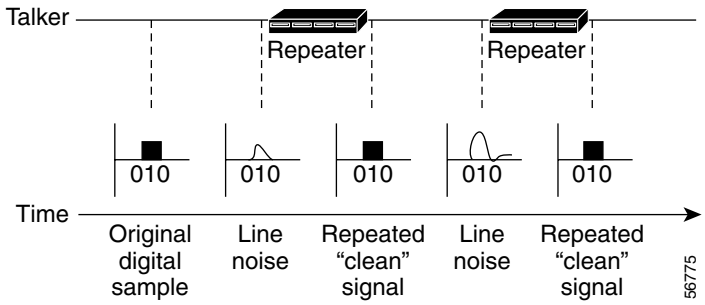
Figure 6 Analog Line Distortion



Digital Signaling

Digital signals are analog signals that are encoded into a stream of binary characters (ones and zeros) using pulse code modulation (PCM). PCM samples the signal and encodes the samples with a fixed number of bits, achieving good-quality voice transmissions. PCM enables “clean sound” or sound without line noise. Processing digital signals is performed using a repeater or digital amplifier. The repeater only has to decide whether to regenerate a binary code of zero or one to achieve clean sound, as shown in Figure 7.

Figure 7 Digital Line Distortion



In digital networks there are many forms of signaling techniques: channel-associated signaling (CAS), R1, R2, ISDN, and Q signaling (QSIG).

Call Control Signaling

Call control is the process of making a routing decision about the destination of a call and making the call happen. In the PSTN today, these call-control decisions are carried out by Signaling System 7 (SS7) and are made by Service Control Points (SCPs).

In a new model of separating the bearers (Real-Time Transport Protocols [RTPs]) from the call-control layer and separating the call-control layer from the services, it is necessary to ensure that standards-based protocols are used. Data networks are unique because multiple protocols can coexist in a network and can be tailored to the particular network needs.

Many different IP routing protocols exist and each is specifically designed for a certain type of network. Each protocol solves a similar routing problem. Each route is slightly different and requires a different tool, which is the routing protocol, to solve the problem. Voice over IP (VoIP) also requires call-control protocols. These protocols all solve the problems of translating phone numbering to IP addresses, but could be used for slightly different purposes.

For instance, H.323 is the most widely deployed VoIP call-control protocol. H.323, however, is not generally seen as a protocol that is robust enough for PSTN networks. For these networks, other protocols such as Media Gateway Control Protocol (MGCP) and Session Initiation Protocol (SIP) are used. No matter what protocol is required, each will be developed to fix a routing problem and serve a particular purpose.

SS7 (Digital PSTN)

SS7 is rapidly replacing multifrequency signaling in the PSTN. By using an overlay network of separate high-speed, out-of-band links operating at 56 or 64 kbps, SS7 signaling reduces network provider expenses for call setup procedures and frees up voice and data trunks to carry the optimal amount of traffic.

SS7 is a method of sending messages between switches for basic call control and for Custom Local Area Signaling Services (CLASS). CLASS is described in the "Features and Services" section on page 33. SS7 also connects switches and databases for network-based services, for example 800-number services and local number portability (LNP).

Some of the benefits of moving to an SS7 network are:

- Reduced post-dialing delay—There is no need to transmit DTMF tones on each hop of the PSTN. The SS7 network transmits all the digits in an initial setup message that includes the entire calling and called number. When in-band signaling is used, each tone normally takes 50 ms to transmit. This means that there is at least a .5-second post-dialing delay per PSTN hop, based on 11-digit dialing.
- Increased call completion—SS7 is an out-of-band signaling protocol, compared to the DTMF or in-band signaling types. SS7 contains all the necessary information (phone numbers, services, and so on) so that the signal is sent faster than tones generated one at a time across an in-band network.
- Connection to the IN—This connection provides new applications and services transparently across switching equipment and speeds up the creation of new services and applications.

Because the SS7 protocol carries the calling number and other critical information with it through the network, it enables sophisticated services to work across an entire network rather than just between subscribers in the same CO. Services, such as these:

- Calling number/calling name display
- Automatic callback
- ISDN networking

VoIP

The central VoIP call-control protocols are H.323, Simple Gateway Control Protocol (SGCP), MGCP, and SIP.

H.323 Protocol

H.323 is the International Telecommunication Union Telecommunication (ITU-T) recommendation that has the largest installed base, simply because it has been around the longest and no other protocol choices existed before H.323. H.323 specifies how multimedia traffic is carried over packet networks. H.323 uses existing standards—for example Q.931—to accomplish its goals. H.323 is a complex protocol that was created not for simple development of applications but to enable multimedia applications to run over unreliable data networks.

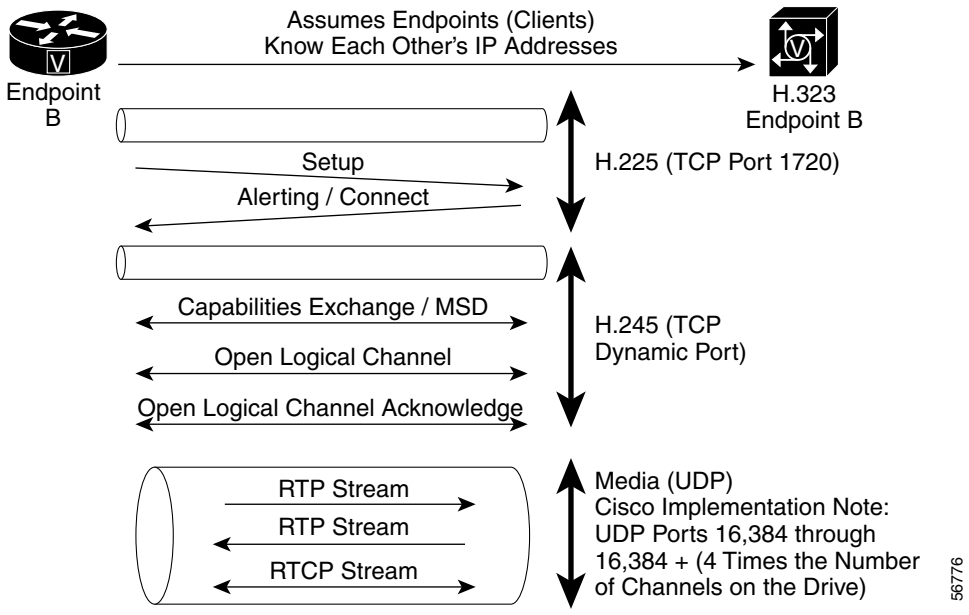
Voice traffic is only one of the applications for H.323. Most of the initial work in this area focused on multimedia applications, with video and data-sharing a major part of the protocol.

Applications require significant work if they are to be scalable with H.323. For example, to accomplish a call transfer requires a separate specification (H.450.2). SGCP and MGCP, on the other hand, can accomplish a call transfer with a simple command, known as a modify connection (MDCX), to the gateway or endpoint. MDCX represents the different approaches built into the protocol design itself—one tailored to large deployment for simple applications (MGCP), and the other tailored to more complicated applications but showing limitations in its scalability (H.323).

To further demonstrate the complexity of H.323, Figure 8 shows a call flow between two H.323 endpoints. This is the most basic H.323 call flow. In most cases, more steps are needed because gatekeepers are involved.

If a call flow is based upon H.323 Version 2, however, H.245 negotiation is enabled in the H.225 setup message. This approach is known as *Fast Start*, which cuts down on the number of round trips required to set up an H.323 call.

Figure 8 H.323 Call Flow



The steps required to complete an H.323 call shown in Figure 8 are as follows:

Step 1 Endpoint A sends a setup message to Endpoint B on TCP Port 1720.

Step 2 Endpoint B replies to the setup message with an alerting message and a port number to start H.245 negotiation.

Step 3 H.245 negotiation includes codec types (G.729 and G.723.1), port numbers for the RTP streams, and notification of other capabilitiesof the endpoints.

Step 4 Logical channels for the UDP stream are negotiated, opened, and acknowledged.

Step 5 Voice is carried over RTP streams.

Step 6 RTTP is used to transmit information about the RTP stream to both endpoints.

SGCP and MGCP

SGCP was developed starting in 1998 to reduce the cost of endpoints (gateways) by having the intelligent call control occur in a centralized platform (or gateway controller). MGCP is basically SGCP with a few additions for operations, administration, maintenance, and provisioning (OAM&P).

SGCP and MGCP were developed to enable a central device, known as a Media Gateway Controller (MGC) or *Soft Switch,* to control endpoints or Media Gateways (MGs). Both of those protocols are referenced simultaneously as xGCP. Applications are developed using standard-based APIs that interface with the MGCs and offer additional functionality (such as call waiting and CLASS features) and applications.

The Cisco version is known as the virtual switch controller (VSC). When the VSC is used, the entire IP network acts like one large virtual switch, with the VSC controlling all the MGs. Figure 9 shows how a typical network design works with a virtual switch running MGCP.

Figure 9 Virtual Switch Controller

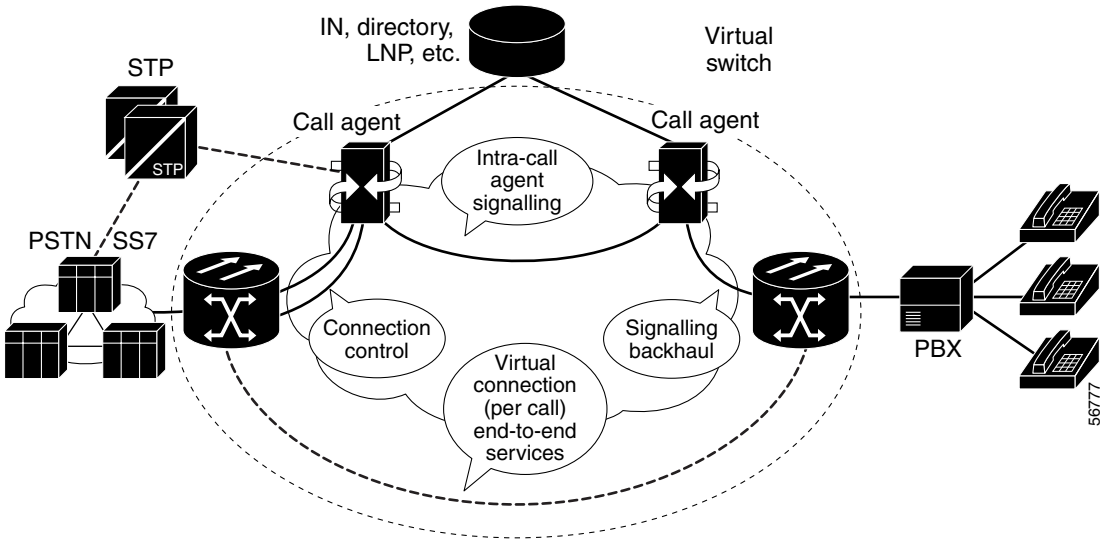


Figure 9 also shows how the legacy PSTN and enterprise networks are connected to gateways or endpoints that enable access into the new packet network. This packet gateway receives direction from the CA, which can communicate with the SS7 network and the IN and can tell the gateways or endpoints how and when to set up a call.

The existing PSTN/SS7 network is connected to the switching transfer point (STP), which also is connected to the MGC or Call Agent. SS7 takes place here. The PSTN/SS7 network is also connected to an MG, which is a signalless trunk that is often known as an Inter-Machine Trunk or IMT. The MG is where the 64-kbps voice trunks are converted into packets and placed onto the IP network.

The MGCs or Call Agents also intercommunicate. Based on the current state of the industry, however, it appears that a variant of SIP or ISDN User Part (ISUP) over IP—a portion of SS7 running on top of IP—will be the primary protocol.

The MGCs have a connection to the IN to provide CLASS services. The MGCs receive signals from the SS7 network and the MGs set up IP connections. The MG on the right side of Figure 9 does not have a connection to the SS7 network. So, a mechanism known as *signaling backhaul* must be used to tell the VSC when and how a call is arriving. Signaling backhaul is normally done with ISDN. The MG or some other device separates the D channel from the B channels and forwards the data or voice information to the MGC through IP.

Session Initiation Protocol (SIP)

SIP is a media-based protocol that enables end devices (endpoints or gateways) to be more intelligent, and enable enhanced services at the call-control layer. SIP is described by RFC 2543, which states that it is an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These multimedia sessions include audio, video, and data and can include multiple partners. SIP enables participants to be invited into an impromptu conference. These multimedia sessions can communicate through multicast, unicast, or a combination of both delivery mechanisms.

Line or Circuit Signaling

There are several kinds of line or circuit signaling, which are described in the following sections:

Loop-Start Signaling

Loop-start signaling is the simplest and least intelligent when compared to ground-start signaling. It also is the most common form of subscriber loop signaling. The protocol works in the same way as the telephone and the local CO, because the creation of a loop initiates a call and the closure of a loop terminates a call. Loop-start signaling is not common for PBXs and has one significant drawback—glare. Glare occurs when two points try to seize the same line, and it often results in two callers being connected unexpectedly. The caller picking up the phone is simultaneously connected to a calling party.

Ground-Start Signaling

Ground-start signaling differs from loop-start signaling, because of the positive recognition of connects and disconnects. Current-detection mechanisms are used at each end of the trunk, enabling end-office switches to agree on which end is seizing the trunk before it is seized. This form of signaling minimizes the effect of glare. It is the preferred signaling method for PBXs.

Channel-Associated Signaling

CAS exists in many networks today. CAS systems carry signaling information in the same channels in which voice and data are carried. Current telecommunication networks require more efficient means of signaling. CAS exists in many varieties that operate over analog and digital facilities. The analog facilities are either two- or four-wire and the digital facilities are either North American T1 or European E1. Each CAS system uses either supervision signaling or address signaling over analog and digital facilities.

Three groups of signals are present in these systems:

- Supervision signals represent events occurring on a trunk and can be specific to CAS. Signal types include seizure, wink, and answer.
- Address signals represent the digits dialed or called party number and, in some instances, other information. Address signals are based on multiflex signaling.
- Tone and announcement signals include ringing and busy tones and announcements specific to an event. Service circuits are used in most exchanges to send and receive address signals and tones as well as to play announcements.

Common Channel Signaling

CCS uses a common link to carry signaling information for a number of trunks. It is cheaper, has faster connect times, and is more flexible than CAS. The first generation of CCS is known as Signaling System 6 (SS6) and the second generation is called, SS7, covered in a previous section.

CCS was originally implemented in 1976 and was called Common Channel Interoffice Signaling (CCIS). CCIS is similar to ITU-T SS6 protocol that operated at low-bit rates (2.4, 4.8, and 9.6 K) and transmitted messages that were only 28 bits in length. CCIS could not adequately support an integrated voice and data environment, so a new standard, CCSS7, was developed.

In-Band Signaling

In-band signaling uses tones instead of direct current (DC) to indicate a change in state. The tones are transmitted over the same facility as voice and are within the 0 to 4 kHz voice band. The tones include:

- Single frequency—Used for interoffice trunks and can be on-hook or idle and off-hook or busy. The single-frequency tone is 2,600 Hz. No tone is present when a connection or circuit is up. When either party hangs up, the 2,600 Hz tone is sent over the circuit, notifying all interoffice exchanges of the disconnect.
- Multi-Frequency (MF)—Used by interoffice trunks to indicate a line seizure, release, answer, acknowledge, and to transmit information, such as the calling party number. MF signaling uses a combination of frequencies across a network. MF signaling is less efficient than common channel signaling (CCS) systems, such as SS7.
- Dual Tone Multi-Frequency (DTMF)—Transmits telephone number digits from the subscriber to the CO. DTMF replaced the transistor oscillators in telephones with keypads and dual-tone oscillators. DTMF tones identify the numbers from 0 to 9 and the asterisk (*) and pound (#) symbols using a combination of frequencies: one from a low group (697, 770, 852, and 941 Hz) and one from a high group (1,290, 1,336, 1,447, and 1,633 Hz). Sixteen possible combinations exist, but only 12 are implemented on the keypad.

Out-of-Band Signaling

Out-of-band signaling uses frequencies or channels outside the frequencies or channels normally used for information transfer. Out-of-band signaling is often used for error reporting in situations in which in-band signaling can be affected by network fluctuations.

Supervisory Signaling

E&M is a common signaling technique used on telephony switches and PBXs as are Foreign Exchange Office (FXO) and Foreign Exchange Station (FXS). For more information about these signaling methods, see the “Configuring Voice Ports” chapter.

Q.931 Signaling

Q.931 is an ITU standard that describes one type of ISDN signaling for Layer 3. The H.225.0 standard uses a variant of Q.931 to establish and disconnect H.323 sessions. This protocols supports user-to-user, circuit-switched, and packet-switched connections. A variety of call-establishment, call-termination, information, and miscellaneous messages are specified, including setup, connect, release, user information, cancel, status, and disconnect.

QSIG Signaling

QSIG is a peer-to-peer signaling system used in corporate voice networking. Internationally, QSIG is known as Private Signaling System No. 1 (PSS1). This open standard is based on the ITU-T Q.9XX series of recommendations for basic service and supplementary services. Therefore, as well as providing inter-PBX communications, QSIG is compatible with public and private ISDN.

QSIG also has one important mechanism known as Generic Functional Procedures (QSIG GF). This mechanism provides a standard method for transporting features transparently across a network.

QSIG has the following functionality:

- It enables the interconnection of multivendor equipment (standards-based protocol).
- It enables inter-PBX basic, feature transparency, and supplementary services.
- It is Interoperable with public and private ISDNs.
- It operates in any network configuration (Star, Mesh, Token Ring, and so on) and is compatible with many PBX-type interfaces.
- It does not impose restrictions on private numbering plans.

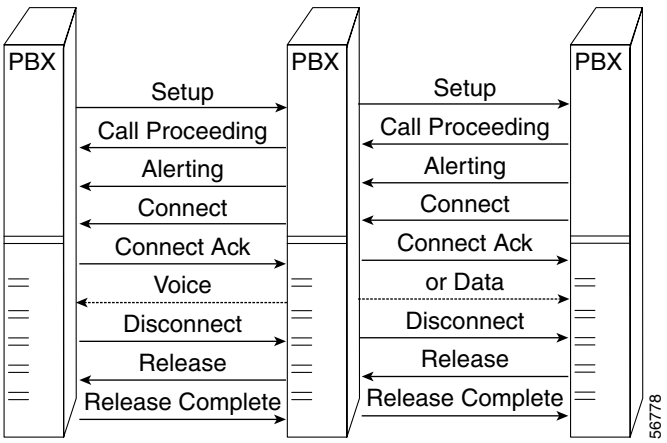
QSIG and ISDN

QSIG end-to-end signaling is maintained from PBX-to-PBX, and ISDN and ISDN User Part (ISUP) interworking is critical for end-to-end signaling in the ISDN network. The ISDN reference model for PBX-to-PBX signaling for corporate networks has two new reference points, Q and C. The reference points are as follows:

- T—Defines access to the NT2 device for ISDN PRI.
- C—Is the physical interconnection point to the PBX. It is compatible with many interfaces, including two- and four-wire analog, BRI, PRI, and radio and satellite links.
- Q—Specifies the logical signaling point between two PBXs. This reference point is used to specify signaling-system and related protocols.

A corporate network (see Figure 10) can have dedicated analog or digital channels, or VPN switched connections. Typically, a T1 or E1 digital interface is used to connect to the network.

Figure 10 Reference Model for Corporate Networks



QSIG Protocols

QSIG protocols specify a signaling system and have an identical structure to that of ISDN, except for layer 3 (shown in Table 5). Layer 3 splits QSIG into the following three sublayers:

- QSIG BC—Interfaces and messages for the user and network sides are identical.
- QSIG GF—Generic layer that enables supplementary services and ANFs and provides a connection-oriented mechanism between the application entities of different PBXs.
- QSIG supplementary service and ANF protocols—Procedures for services and ANFs that are defined by the European Computer Manufacturers Association (ECMA) and the European Telecommunication Standards Institute (ETSI).

Table 5 QSIG Protocols

OSI Reference	QSIG Protocol	QSIG Standard
L1	None	Based on used interface.
L2	None	Identical to ISDN L2 (LAPD[1]).
L3	QSIG BC	ECMA 142/143; ETS300[2] 171/172.
	QSIG GF	ECMA 165; ETS300 239.
	QSIG (supplementary services)	Separate specifications, such as call forward (ECMA 173/174, ETS300 256/257) and call transfer (ECMA 177/178, ETS300 260/261).
L4	Application-based service elements.	Transparent to the network.
L5		
L6		
L7		

1. Link Access Procedure on the D channel.
2. ETSI-based standard.

ISDN

ISDN is a network that can consist of T1, T3, E1, and E3 and has two types of subscriber access: Basic Rate Interface (BRI) and Primary Rate Interface (PRI). Each access is comprised of B and D channels. B channels are 64-kbps channels that carry user information streams. No signaling information is carried in the B channel. The user streams include speech encoded at 64 kbps according to ITU G.711, data at or less than 64 kbps, and voice encoded at lower bit rates.

D channels are used primarily to carry signaling for circuit switching by ISDN networks. D-channel bit rates are different depending on the access method. The D channel also is capable of transmitting user packet data up to 9.6 kbps.

An in-depth description of each type is as follows:

- BRI—Useful when Digital Subscriber Lines (DSLs) or cable modems were unavailable and provided a fast connection to the Internet. Delivers two bi-directional 64-kbps B channels and one bi-directional 16-kbps D channel over standard two-wire telephone lines. Basic rate ISDN service typically is used for residential and small office, home office (SOHO) applications. Each B channel can transmit speech or data; the D channel transmits the signaling or call control messages. The reference configuration for ISDN is defined in the ITU specification I.411. The reference points specify the transmission medium, interface, and connectors (if used).
  - U reference point—Specifies the transmission characteristics of the local loop. The two-wire interface operates at 160 kbps (2B+D + 16 kbps for overhead) over standard copper twisted wires.
  - S/T reference point—Provides a four-wire connection to ISDN-compatible terminals or terminal adapters. The interface operates at 144 kbps (2B+D) between the ISDN device and the network termination device. Up to eight ISDN devices can be connected to the S/T interface.
  - R reference point—Provides connection using EIA/TIA-232 and V.35 interfaces for non-ISDN devices. The devices connect to the terminal adapter. This reference configuration also specifies the set of functions required to access ISDN networks:
    - Network Termination 1 (NT1)—Outside the United States, NT1 is on the network side of the defined user-network interface and is considered part of the service provider network. NT1s terminate the two-wire local loop and provide four-wire S/T bus for ISDN terminal equipment (TE).
    - Terminal Equipment 1 (TE1)—ISDN-compatible devices that connect directly to the S/T connector on the NT1.
    - TE2—Non-ISDN compatible devices that require terminal adapter (TA) interconnection.
    - TA—An ISDN-compliant interface to NT1s and standard interfaces for TE2s. These standard interfaces include EIA/TIA-232, V.35, EIA/TIA-449, and X.21.
- PRI—Designed for telephone switches, computer telephony, and voice processing systems. PRI can be made into as many as 24 and 32 phone calls.Corresponds to two primary rates: 1.544 Mbps (T1) and 2.048 Mbps (E1). PRIs typically are used in medium to large business applications. PRI is comprised of 23 64-kbps B channels and one D channel. The interface structure for T1 is 23B + D (North America and Japan). The interface structure for E1 is 30B + D (Europe). The configuration and reference points for PRI are similar to those for BRI, and the differences:
  - U reference point—A four-wire interface that operates at T1 PRI rates.
  - T reference point—Provides access to the Network Termination 2 (NT2) device.
  - NT2—PBX equipment can provide such NT2 functions as Layer 2 (L2) and Layer 3 (L3) protocol handling as well as multiplexing, switching, interface termination, and maintenance. NT2s also can provide connections to ISDN-compatible TE1s and non-ISDN compatible TE2s.

ISDN was designed to overcome the deficiencies of the PSTN by the following:

- Providing an internally accepted standard for voice, data, and signaling.
- Making all transmission circuits end-to-end digital.
- Adopting a standard out-of-band signaling system.
- Bringing significantly more bandwidth to the desktop.

Some ISDN features are as follows:

- Call waiting—The calling party is placed in a queue until the called party accepts or rejects the call.
- Specialized numbering and dialing plans—Centralized management of all ISDN terminals, including PBXs, key systems, and so on.
- Credit card calling—Automatic billing of certain or all calls into accounts independent of the calling line.
- Calling line identification presentation—Provides the calling party the ISDN phone number and address, in some cases, of the called party. The called party can accept or reject the call.
- Calling line identification restriction—Restricts presentation of the calling party's ISDN phone number and address to the called party.
- Closed user group—Restricts conversations to or among a select group of phone numbers, local, long distance, or international.
- Desktop videoconferencing—Enables the display of the calling party's video image on the desktop device.
- E-mail—ISDN can carry information to and from unattended phones as long as the phones are equipped with the required hardware and software.
- Simultaneous data calls—Two users can talk and exchange information over the D packet and the B circuit.

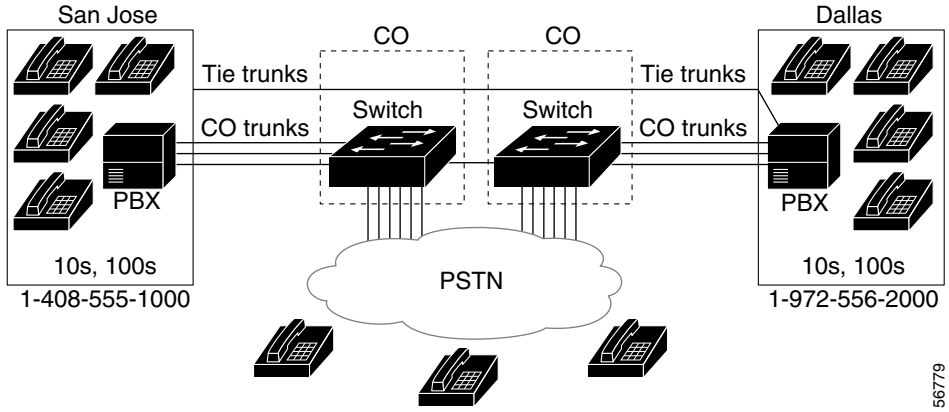
Trunking

The terms line and trunk are often used synonymously, but there is a distinction. A trunk can connect key systems in private telephone systems, routers, and switches. A line connects to a PBX, switch, or other communication system, telephone, computer terminal, or other endpoint. There are several kinds of trunks, such as tie, loop-start, ground-start, and ISDN (T1, E1, BRI, and PRI). Some use analog signaling and others use digital signaling.

Tie-Line Trunks

If a business has two sites with a large call volume between the sites, the business can purchase a tie-line trunk. A tie-line is a permanent circuit or trunk between two points (T1, E1, fractional T1/E1, or some other transport). Figure 11 shows two sites (one in San Jose, California, and one in Dallas, Texas) connected by a T1 circuit.

Figure 11 Tie-Line Between San Jose and Dallas



The tie-line uses the PSTN, but the business pays a flat rate for the dedicated use of the trunk between San Jose and Dallas. The PBX uses a pre-programmed Automated Route Selection (ARS) table to determine which trunk should be used. In this case, the PBX is configured to use the tie-line trunk between San Jose and Dallas. If the tie-line is full, the PBX uses the CO trunks as overflow to the PSTN.

Analog Trunks

A single-frequency 2600 Hz tone indicates the use of analog signaling. The tone is applied in-band over a trunk and is turned off when a call is in progress or established. The trunk or line is on-hook or idle when tone is present and off-hook or in use when tone is absent.

Suppose that Switch A sends forward signals and Switch B sends the backward signals. Switch A sends a forward seizure or off-hook signal to Switch B on a chosen trunk. Then Switch B sends a backward wink or proceed-to-send to Switch A and waits for address signaling or dialed digits.

After the digits are sent and the call is answered, Switch B sends a backward answer or off-hook to Switch A, enabling an end-to-end voice path. In this case, the calling party hangs up first and a clear-forward signal is sent from Switch A to Switch B. When the called party hangs up, a clear-back signal is sent by Switch B.

Digital Trunks

In businesses the most commonly used trunks today are digital trunks in T1 or E1 facilities. With digital trunks, bits are robbed from specific frames and are used for signaling purposes. T1 has two types of framing formats:

- Super Frame (SF)—Least significant bits are robbed from frames 6 and 12.
- Extended Superframe (ESF)—Least significant bits are robbed from frames 6, 12, 18, and 24.

The SF signaling bits are equal to each other and provide two-state, continuous supervision signaling. Bit values of zero are used to indicate on-hook, and bit values of 1 are used to indicate off-hook.

With the introduction of the digital E1 packet voice, network modules can connect to a PBX (or similar telephony device) or to a CO in order to provide PSTN connectivity. The differences that set E1 digital configuration apart from analog configuration are as follows:

- Timing—Analog interfaces do not require specific timing configuration. Digital E1 interfaces require not only that you set timing but that you consider the source of the timers.
- Framing—Analog interfaces do not require specific framing configuration. Digital E1 interfaces require that you configure for cyclic redundancy checking 4 (CRC-4) framing. Set the framing format to match that of the PBX or CO that connects to the digital E1 packet voice trunk network module.
- Line encoding—Analog interfaces do not require specific line encoding configuration. Digital E1 interfaces require that you configure for high density binary 3 (HDB3) encoding (similar to alternate mark inversion, or AMI). Set the line encoding to match that of the PBX or CO that connects to the digital E1 packet voice trunk network module.

T1/E1 Trunks

T1 trunks are digital transmission links with a total signaling speed of 1.544 Mbps. T1 is the standard for North America, and E1 is the European standard. E1 carries data at the rate of 2.048 million bits per second (DS-1 level) and designed to carry 32 64-kbps digital channels.

T1 is part of a progression of digital transmission pipes—a hierarchy known generically as the digital signal (DS) level hierarchy. T1 used to be delivered on two pairs of unshielded twisted copper wires—one for sending and one for receiving. The combination of these two unidirectional (simplex) circuits yields a bidirectional (full duplex) circuit. Now, T1 is often delivered on fiber-optic transmission systems by the CLECs and ILECs where fiber is available. T1 lines can be leased as a channelized service and delivered as separate voice or data channels or as an unchannelized raw bit stream.

Typically in North America, channelized T1, split into 24 voice-grade channels with each running at 56 kbps, is used for voice and unchannelized is used for data. Unchannelized is more appropriate for compressed voice, video, and IP telephony because 1.546 Mbps is provided and can be split any way that is required.

T3/E3 Trunks

T3 is the North American standard for DS-3. T3 operates at a signaling rate of 44.736 Mbps, equivalent to 28 T1s. T3 is capable of handling 672 voice conversations, each at 64 kbps, and runs on fiber optic or microwave transmission.

E3 is a wide-area digital transmission scheme that is used predominantly in Europe and that carries data at a rate of 34.368 Mbps. E3 lines can be leased for private use from common carriers.

Dial Plans

A dial plan is a method of assigning individual or blocks of telephone numbers (E.164 addresses) to physical lines or circuits. In the PSTN, a dial plan is created by partitioning blocks of numbers in a hierarchy (10,000 numbers is normal for the PSTN).

To create a dial plan for an enterprise voice network, individual telephone numbers can also be assigned to individual users. Even in private enterprise voice networks, it is common to adopt hierarchical assignments when creating a dial plan. Although dial plans in the PSTN are not simple, they are at least hierarchical and enable hierarchical dial plans.

For more information on dial plans, see the "Configuring Dial Peers, Dial Plans, and Digit Manipulation" chapter in this guide.

Cisco Dial-Plan Implementation for VoIP Network

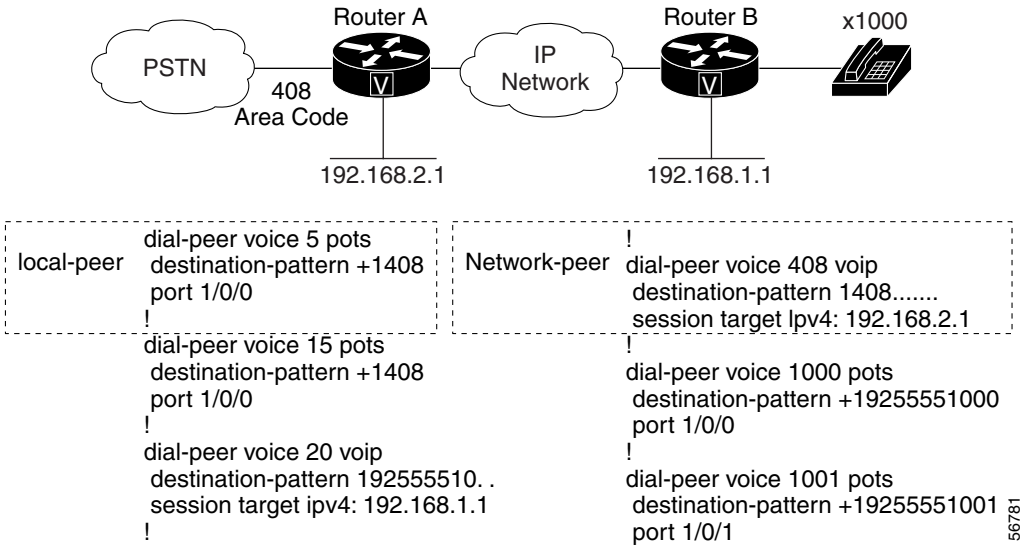
A fundamental VoIP network must have the following features:

- Local dial peers to map phone numbers to a physical port
- Network dial peers to map phone numbers to an IP address
- The ability to strip and add digits
- Number expansion

The dial peer is a concept that enables all these basic features. A dial peer exists as a local (PSTN) and network (VoIP*)* dial peer. To route a call more efficiently, network managers can add, replace, or reduce the number of dialed digits, which is a procedure called number expansion. This procedure also enables overlapping dial plans to coexist. Local dial peers strip away all digits matching a specific substring noted in the destination-pattern command.

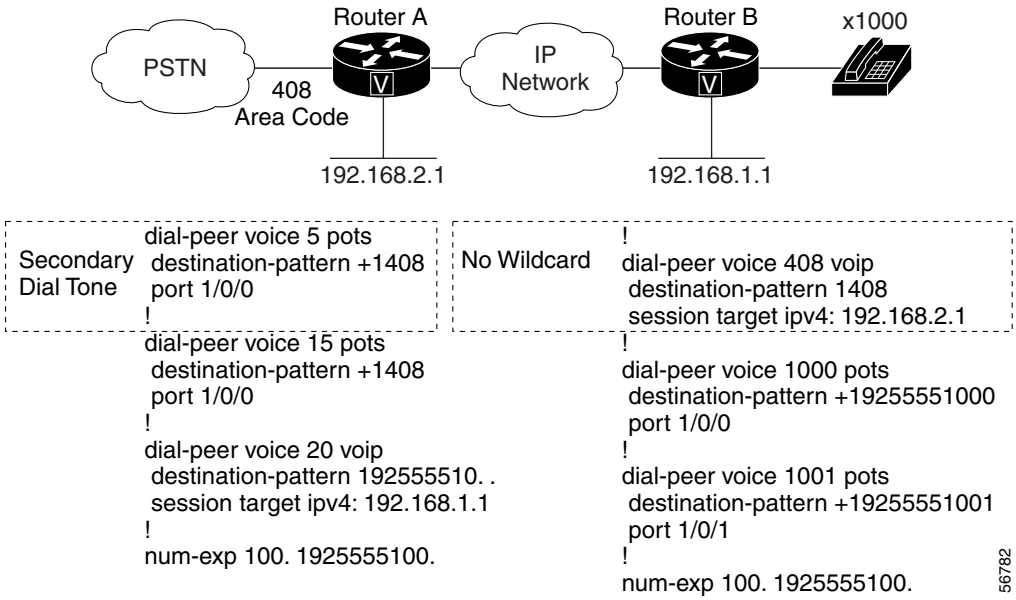
Figure 12 shows that dial peer 5 is a local peer because it is labeled "pots" and port 1/0/0 on the Router A is mapped to the phone number "1408". Router B has a network peer, denoted by the "voip" label, which maps the phone number "1408....." to an IP address. The periods act as wildcards. The seven periods tell the phone to wait until "1408" and 7 more digits are received before attempting to complete the call. If the voice gateway receives 1408 plus seven digits, it matches that dial peer statement and attempts to connect to the session target.

Figure 12 Cisco Dial-Plan Configuration



Another possibility is to use two-stage dialing. Figure 13 shows the changes to the dial plan when two-stage dialing is implemented.

Figure 13 Cisco Two-Stage Dialing Configuration



Router A has no wildcards in Figure 13 because the Router A strips all the digits based on **dial-peer voice 5 pots** command and only a secondary dial tone is sent back to the user at extension 1000 on Router B. This enables the user at extension 1000 to dial any location because the dial plan on either router is no longer used.

The following features also are available on Cisco routers:

- Hunt groups—Enable the voice gateway to hunt through dial peers to ensure delivery of a phone call to a valid IP gateway.
- Call failover—Enables an IP call to be routed to a different location if the first IP destination is unreachable.
- Busyout—Enables the gateway to set the physical voice-signaling port to “busy” when network congestion or network failure occurs.
- Trunking—Enables two VoIP gateways to act as a tie-line (both digital tie-lines and analog tie-lines are supported).

For more information about dial peers, see the “Configuring Dial Plans, Dial Peers, and Digit Manipulation” chapter in this guide.

Features and Services

Some of the available features and services are described in the following sections:

CLASS Features and Services

CLASS consists of number-translation services, such as call forwarding and caller identification, available within a local exchange of a LATA. Some of the CLASS phone services are as follows:

- Automatic callback—Notifies the user when the called line is not busy by placing the call.
- Automatic recall—Enables the user to easily return a missed call.
- Calling number blocking—Enables the called party to hide its number and name from the caller.
- Customer originated trace—Enables the user to dial a code after receiving a harassing call, thereby notifying the local law enforcement agency.
- Call screening—Enables the user to accept, reject, or forward calls based on a list of received calling numbers.

CLASS features provide customers with a powerful and convenient tool to control incoming and outgoing calls. CLASS enables users to interact with the switch software from their own telephone sets and give instructions on which services they want. SS7 messages and functions are then invoked and sent within the network to perform the requested operations.

QSIG Features and Services

QSIG supports a suite of features and services for corporate PBX networks. The three main service groups are basic services, generic functional procedures, and supplementary services.

- Basic service (QSIG BC)—Provides set up, manage, and tear down of a call. Similar to an ISDN bearer service, basic services include speech, 3.1-kHz audio, and 64-kbps unrestricted.
- QSIG GF—Transports nonstandard features using a standardized method, thus providing feature transparency. This mechanism enables the exchange of signaling information for the control of supplementary and additional network features over a corporate network.
- Supplementary services—Includes services as well as additional network features (ANFs). Supplementary services and ANFs include call completion, call forward, call diversion, call transfer, call waiting, caller ID, and advice of charge.

Debit Card Feature

The Debit Card for Packet Telephony feature works in conjunction with the Cisco interactive voice response (IVR) software, authentication, authorization, and accounting (AAA), and RADIUS, and with an integrated third-party billing system. The IVR software infrastructure allows prerecorded audio files to be combined dynamically to play the dollar amount of credit remaining, the time and date, and other information.

The integrated third-party billing system maintains per-user credit balance information. The Debit Card for Packet Telephony feature uses AAA and RADIUS vendor-specific attributes (VSAs) to communicate with the billing system. The Debit Card for Packet Telephony feature includes the ability to maintain per-user credit balance information through the use of a billing system. When these features are implemented, the billing system and Cisco IOS software functions enable a carrier to authorize voice calls and debit individual user accounts in real time at the edges of a VoIP network without requiring external service nodes.

The Debit Card for Packet Telephony feature includes the following functionality:

- Rates a call according to the caller ID, personal identification number (PIN), and destination number.
- Plays the credit (dollar amount) remaining on a card in $$$$$$.$$ format.
- Announces the time remaining credit on the card in hours and minutes (HH:MM).
- Plays a “time-running-out” message based on the configurable time-out value.
- Plays a warning “time-has-run-out” message when the credit runs out.
- Makes more than one successive call to different destinations during a single call session.
- Reauthorizes each new call.
- Allows type-ahead keypad entries without waiting for the prompt to complete.
- Allows the caller to skip past announcements by pressing a touch tone key.
- Allows retry when entering data (user ID/PIN/destination number) by using a special key.
- Terminates a field by size rather than by using the terminating character (#).
- Supports two languages.
- Sends an off-net tone to the caller.
- Provides voice-quality information to the RADIUS server on a call-by-call basis.
- Uses prompt memory more efficiently.
- Creates dynamic prompts by using prerecorded audio files.
- Allows retries for RADIUS server failures, with the maximum number retries allowed determined by the RADIUS server.

Telephony Applications

Some of the available telephony applications are described in the following sections:

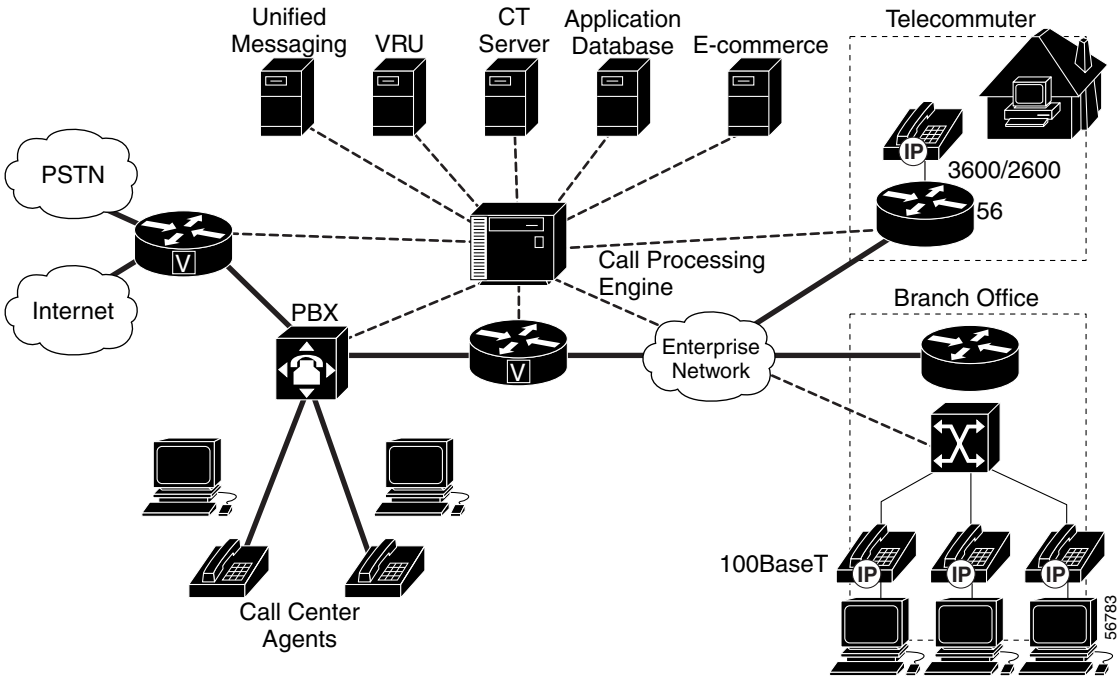
- Call Center Applications, page 35
- Cisco AAVID Multiservice Network, page 38
- Computer Telephony Integration, page 38

Call Center Applications

The call center infrastructure and the applications that drive the centers grew out of a need to cut costs (building rent, infrastructure costs, and so on) and provide customers with efficient and timely service. In most call centers today, the largest costs are for the brick and mortar holding the building together. There are two types of call centers: Packet Telephony Call Center (PTCC) and Circuit-Switching Call Center (CSCC).

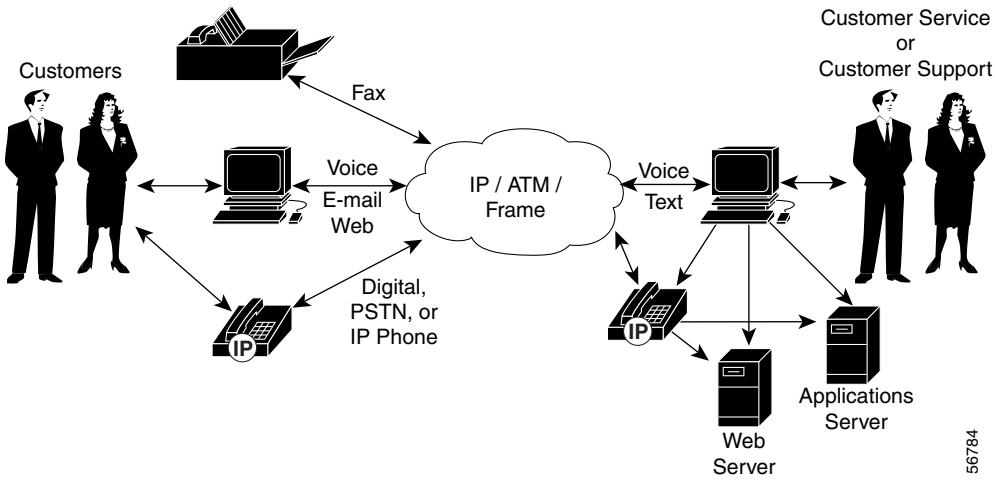
A PTCC enables a connection into a legacy PBX call center and integration into web support, Internet telephony, and unified communications (see Figure 14).

Figure 14 Packet Telephony Call Center



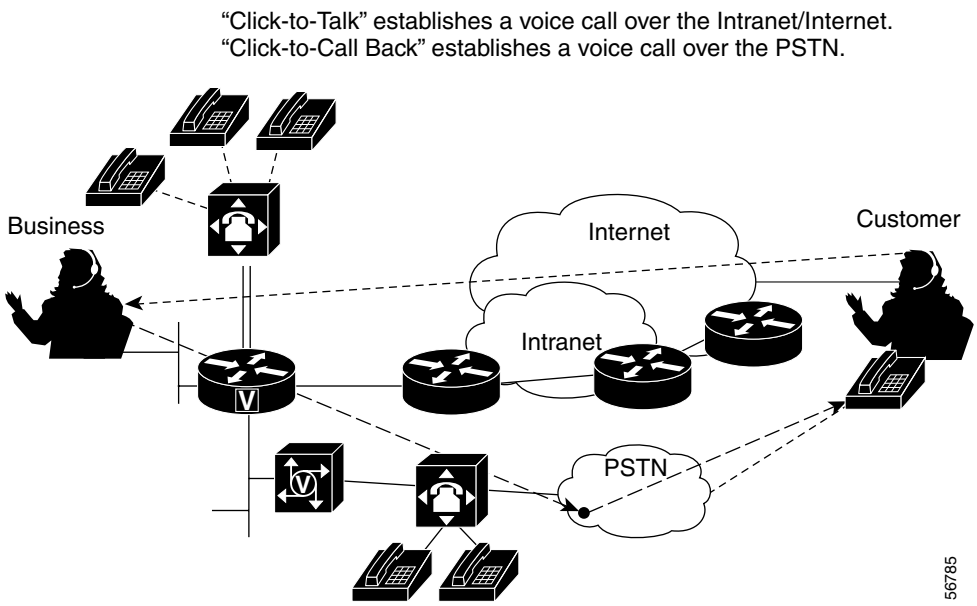
The call center is no longer tied to physical ports, but is tied into one common messaging infrastructure (e-mail, voice mail, and so on). The call-routing or call-processing engine is now part of the data network and is removed from the PBX. This structure enables telecommuters, call-center agents, and branch office agents to have the same access to the same information. A common infrastructure gives the call agent a customer profile that has a common look and feel, as shown in Figure 15.

Figure 15 Common Infrastructure for All Call Agents



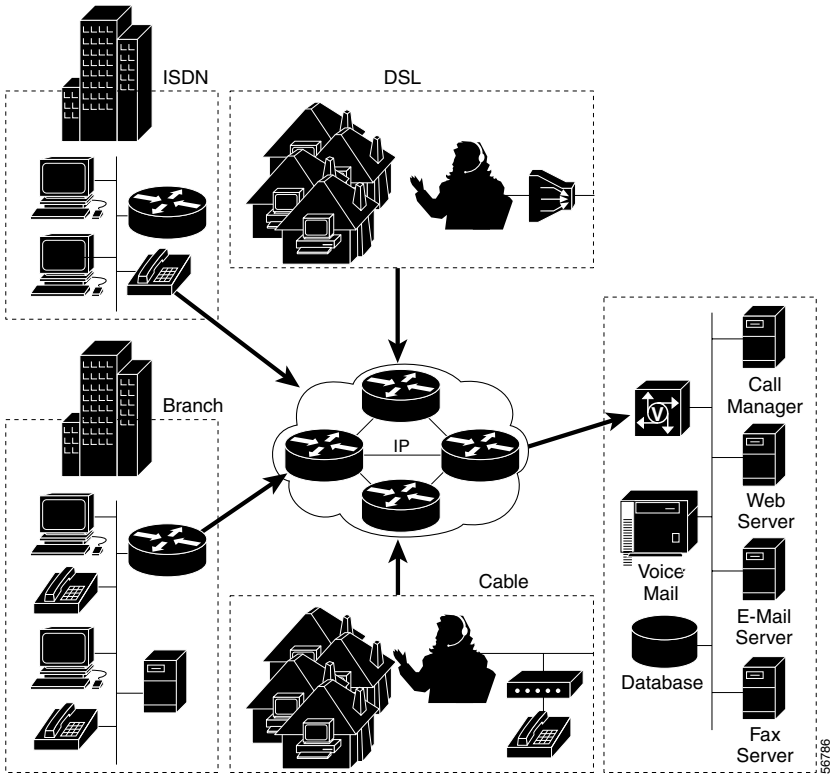
This new architecture uses the existing WAN data infrastructure and provides a more efficient use of existing bandwidth. Another benefit is web integration, which means that a call-center customer can request a call back from the website. The call center routes the customers to the proper agent depending upon where they click. This application (shown in Figure 16) is known as click-to-talk or click-to-dial.

Figure 16 Click-to-Talk



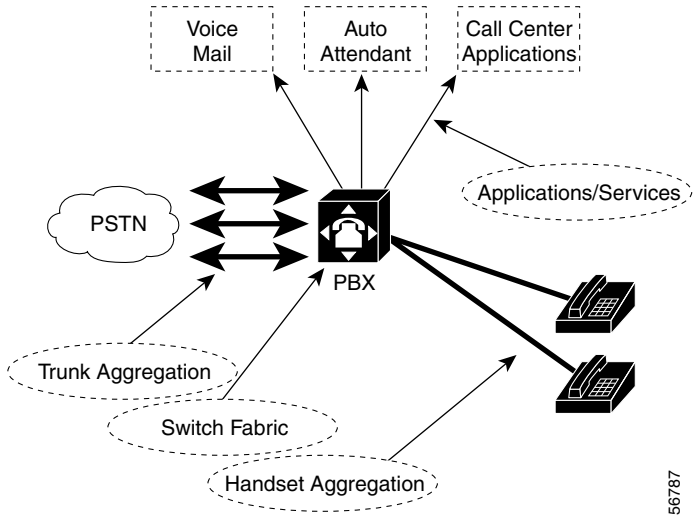
In a PTCC infrastructure that uses a VoIP network, there can be a group of distributed virtual agents that can be located anywhere and have the same tools that a traditional call center offers. Figure 17 shows a virtual agent architecture.

Figure 17 Virtual Agents



In the CSCC environment, shown in Figure 18, a remote piece of equipment extends the features of the PBX to the user premises.

Figure 18 Circuit-Switching Call Center



Cisco AAVID Multiservice Network

Cisco Architecture for Voice, Video, and Integrated Data (AAVID) meets the requirements of the multiservice network. Cisco AVVID has a five-phase multiservice strategy, providing an architecture for converged networking.

Cisco AVVID integrates all communications into a single infrastructure, providing a single point of management, administration, and control. Its ability to accommodate integrated IP-based applications improves individual and group productivity while providing for highly personalized user and customer experiences. And the open architecture and ease of scalability of Cisco AVVID makes it a highly adaptable multiservice network solution that can be implemented in ways that leverage existing network investments in legacy equipment and solutions.

The Cisco AVVID architecture consists of three distinct building blocks that combine to provide a complete IP-based end-to-end solution for meeting the needs of today's emerging multiservice networks. The most important building block is the Cisco AVVID infrastructure: the routers, switches, cache engines, and gateways over which the Cisco IP fabric of intelligent network services run.

Next are the clients, which include an array of IP telephones, SoftPhones, PCs, and video equipment. A Cisco IP SoftPhone running on a PC can control a Cisco IP phone, bringing the power of the PC user interface to telephony without the need for complex computer telephony integration programming. And the third building block of Cisco AVVID comprises a range of multiservice network applications—from Cisco and from its many partners—enabled by an expanding variety of servers required to distribute these applications across the network.

Computer Telephony Integration

To meet service demands, Computer Telephony Integration (CTI) was developed. CTI is many applications that use a PC. For example, one CTI application displays caller information (such as name, buying patterns, and address) by "popping" the information on the agent screen so that the agent can handle the call more quickly. Another application routes calls to the proper agent based upon technical skills, language, and any other skill to further increase the speed by which the call is handled. IVR is another application that enables callers to input basic information (such as account information) so that calls can be handled more quickly.

IVR consists of simple voice prompting and digit collection to gather caller information for authenticating the user and identifying the destination. IVR applications can be assigned to specific ports or invoked on the basis of DNIS. An IP PSTN gateway can have several customized IVR applications to accommodate many different gateway services. The customized IVR applications can present different interfaces to the various callers.

IVR systems provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words, or more commonly dual tone multifrequency (DTMF) signaling. For example, when a user makes a call with a debit card, an IVR application is used to prompt the caller to enter a specific type of information, such as an account number. After playing the voice prompt, the IVR application collects the predetermined number of touch tones and then places the call to the destination phone or system.

TCL IVR uses Tool Command Language (TCL) scripts to gather information and to process accounting and billing. For example, a TCL IVR script plays when a caller receives a voice-prompt instruction to enter a specific type of information, such as a personal identification number. After playing the voice prompt, the TCL IVR application collects the predetermined number of touch tones and sends the collected information to an external server for user authentication and authorization.

Some services and applications include integrated voice mail and e-mail into one application, using web-based customer support, having CTI ability. Some also enable the user to fax from the desktop and to an e-mail account. Still other applications enable the user to conduct desktop video conferencing with the customer.

Enterprise Telephony

Enterprise telephony (ET) provides basic business features, such as hold, three-way calling, call transfer, and call forwarding ET provides the following functionality:

- Circuit switching based upon 64-kbps circuit switching
- Infrastructure model with bearers, call-control, and service planes contained in one platform
- Local loop in which phones can plug directly into a switch and receive a dial tone, place and receive phone calls, and so on

ET switches 64-kbps circuits and is equivalent to a Class 5 switch (PBX), supporting from five to several thousand local loops.

A Class 5 switch provides residential telephony, with a few basic business features, such as call waiting and call return. A PBX usually has many more features, including call hold, three-way calling, call transfer, and voice mail.

PSTN and ET are different from each other in the way they treat signaling and in the types of features they offer. Although the PSTN uses signaling interfaces developed by the industry, PBX manufacturers often create proprietary protocols to enable intercommunication and additional features transparently throughout the voice network.

SS7, ISDN, and in-band signaling are the primary signaling links. Many PBXs in ET use CAS and PRI for signaling. In many cases, CTI links enable a third-party computer application to control some of PBX operations. Many vendors are starting to implement standards-based signaling protocols that enable interoperability between different PBXs. The most common protocol is QSIG.

Providing advanced features is also an important differentiation between ET and PSTN. Business requirements for telephone networks are much greater than the average home user. ET customers have the need for high-use, feature-rich systems that enable applications such as:

- Inbound and outbound call centers—ET networks with this feature usually contain a CTI link that enables new applications. For example, a screen pop on the agent computer screen that gives the agent caller-ID information as well as caller buying habits and shipping address.
- Financial Enterprise Telephony—ET networks with this feature often include a network known as hoot-n-holler, in which one person speaks and many people listen. This is common in stock brokerage offices.

ET customers can use the PSTN to service basic PBX needs, but the PSTN does not have advanced applications such as call centers. Also, using PSTN is usually more costly than using ET, and the PSTN might not have all the necessary functionality that the enterprise customer needs.

Cisco IP Telephony

By using the Cisco CallManager, a PBX can be eliminated and replaced with IP telephony over a converged network. The Cisco CallManager provides call-control functionality and, when used in conjunction with the IP telephone sets or a soft telephone application, can provide the PBX functionality in a distributed and scalable fashion. Cisco CallManagers can be networked via IP and provide fall back to the PSTN if required.

Today users have a wide range of communication and messaging media available to them: telephones, cell phones, pagers, fax, voice mail, and e-mail. Each of these requires distinct hardware and software components to function. Unified messaging combines voice mail, e-mail, and fax into a single application suite.

With unified messaging, a single application can be used to store and retrieve an entire suite of message types. Voice-mail messages stored as Windows audio video (WAV) files can be downloaded as e-mail attachments while traveling, a response recorded and returned to the sender, all recipients, or an expanded list. E-mail can be retrieved via a telephony user interface (TUI), converted from text to speech, and reviewed from an airport lobby phone or a cell phone. Infrastructure is decreased as now a single application can provide voice, e-mail, and fax. Productivity is increased because what were once disparate message types can be retrieved via the user's preferred interface.

Common ET Designs

ET designs generally consist of an interface between PSTN and the enterprise network. The interface can be as simple as an analog line from the PSTN or a leased line between two PBXs. Or, it can be as complex as an Asynchronous Transfer Mode (ATM) connection using an IXC public ATM network.

There are five designs that businesses can choose, each of which uses slightly different components which are:

- Simple business line—This design is one a line coming directly from the PSTN. The line is similar to a residential line, however, the monthly rate is higher. The service is provided and managed by the LEC or Competitive LEC (CLEC).
- Key system—A key system is generally used in offices of fewer than 50 people.
- PBX—A PBX provides many features (such as hold, transfer, park, and so on) that business customers require. This switch often connects to the PSTN through a T1 or E1 circuit and often integrate voice mail, local lines, and PSTN trunks.
- Centrex line—This line is provided and managed by the LEC or CLEC and offers additional services similar to a PBX, with an additional monthly charge. The services include transfer, three-way calling, and a closed user-dialing plan.
- Virtual Private Networks (VPNs)—With a VPN, the PSTN contains a private dial plan for the enterprise customer. LECs, CLECs, and IXCs can provide VPNs. A local PBX can provide additional features.

Cisco Voice Technologies and Concepts

This section discusses the following technologies and concepts:

Voice over IP

VoIP is a feature for service-based networks and is essentially samples of analog information. VoIP is deployed in a number of ways, including "soft" phone or intranet-attached handsets and off-the-network calling systems that move voice traffic across data networks.

Handling voice in a best-effort environment builds upon QoS services, but extends QoS with some specific capabilities:

- Compression/decompression (codec) algorithms squeeze voice traffic into a fraction of the space that it uses in traditional circuit-based telephone systems.
- A gateway function connects legacy systems to the IP telephony environment.
- Call management features (Cisco Call Manager) deliver the functionality of the phone system, such as call forwarding, three-way calling, dial tone, call routing, transfer, and so on.
- Programming interfaces enable applications to communicate with the voice services infrastructure and to request call management changes. In this way, a help desk system might request that the voice service transfer a call to a particular extension where the most qualified recipient can handle a customer issue.

VoIP offers compelling benefits to the business environment because of the reuse of existing networks, reduced support costs, opportunity for cheaper call paths and less long distance spending, and integration of other voice applications like faxing.

The main business motivators for deploying voice services are budget leverage (achieved by consolidating voice and data infrastructures) and the improved competitiveness of new applications that increase productivity and enhance the customer experience. Voice services permit the deployment of computer-telephony integration (CTI) systems for voicemail, unified messaging, help desk, and customer service operations.

For detailed information, see the "Voice over IP Overview" chapter in this guide.

Voice over Frame Relay

Voice over Frame Relay (VoFR) enables a router to carry voice traffic (for example, telephone calls and faxes) over a Frame Relay network, using the FRF.11 protocol. The FRF.11 specification defines multiplexed data, voice, fax, DTMF digit-relay, and CAS/robbed-bit signaling frame formats. The Frame Relay backbone must be configured to include the map class and Local Management Interface (LMI).

The Cisco VoFR implementation enables dynamic- and tandem-switched calls and Cisco-trunk (private-line) calls. Dynamic-switched calls have dial-plan information included that processes and routes calls based on the telephone numbers. The dial-plan information is contained within dial peer entries.

For detailed information and configuration task tables, see the "Configuring Voice over Frame Relay" chapter in this guide.

Voice over ATM

Voice over ATM (VoATM) enables a router to carry voice traffic (for example, telephone calls and faxes) over an ATM network. An ATM network is a cell-switching and multiplexing technology designed to combine the benefits of circuit switching (constant transmission delay and guaranteed capacity) and packet switching (flexibility and efficiency for intermittent traffic).

An ATM connection transfers raw bits of information to a destination router or host. The ATM router takes the common part convergence sublayer (CPCS) frame, carves it up into 53-byte cells, and sends the cells to the destination router or host for reassembly.

For detailed information and configuration task tables, see the "Configuring Voice over ATM" chapter in this guide.

Multimedia Conference Manager (MCM)

MCM is a feature set that enables IP networks to support secure, reliable videoconferencing, with advanced QoS capabilities. MCM functions as a high-performance H.323 gatekeeper and proxy, enabling network managers to control bandwidth and priority setting for videoconferencing services based on individual network configurations and capacities. These capabilities ensure appropriate allocation of network resources for conferencing as well as other critical applications running simultaneously on the network. MCM can scale to accommodate small, medium-sized, or large conferencing environments.

MCM differentiates itself from other H.323 gatekeepers because it:

- Offers proxy services as well as gatekeeper services.
- Combines gatekeeper/proxy services with routing capabilities on a single hardware platform.
- Supports a multiservice IP networking environment for data, voice, and videoconferencing on a common software base.
- Offers scalability by virtue of its availability on a wide range of platforms.
- Offers excellent price/performance for small to very large H.323 network deployments.

Those capabilities make the MCM an important component of the Cisco AVVID which integrates all of these communications into a single multiservice IP network infrastructure. The MCM gatekeeper and proxy provide IP conferencing networks with features such as:

- Packet forwarding services
- Address resolution and call routing
- User authentication and call accounting

- Bandwidth management
- QoS connection signaling

These features enable conference users to experience high quality, even when other applications are running on the network.

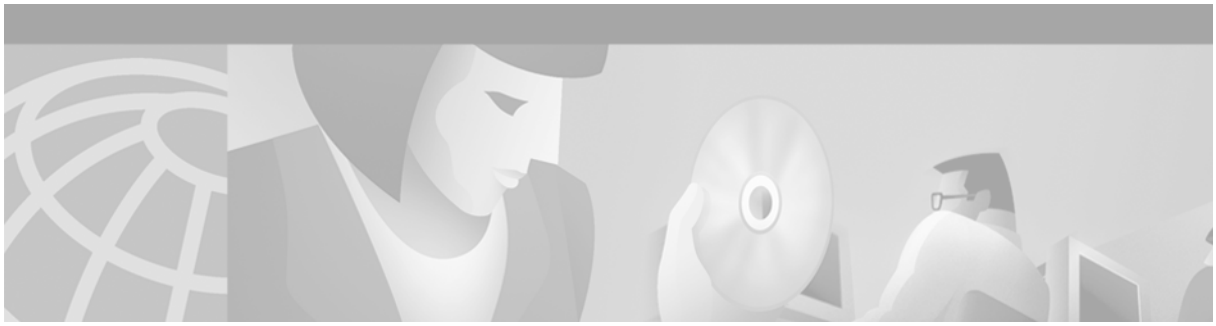
Fax Applications

Fax applications enable the sending and receiving of faxes across packet-based networks using modems or voice feature cards (VFCs). Some of the benefits of the applications are as follows:

- Universal inbox for fax and e-mail—Faxes and e-mails can go to the same mailbox using direct inward dialing (DID) numbers. E-mail and fax recipients can be combined.
- Toll bypass—In an enterprise environment in which offices in different cities are connected using a WAN, toll charges can be bypassed by transmitting faxes over the network connection. Because a fax message is stored on the mail server until Simple Mail Transfer Protocol (SMTP) forwards messages to the recipient, SMTP can forward fax e-mail attachments during off-peak hours (for example, during evenings and weekends), thereby reducing long-distance charges.
- Broadcast to multiple recipients—E-mail fax attachments can be sent to multiple recipients simultaneously.
- Cost savings and port density using T.37/T.38 Fax Gateway—The cost of maintaining one architecture (either fax or voice) is eliminated.

Quality of Service

The basic goal of quality of service (QoS) is to maximize bandwidth and latency for a specific application. The network administrator can group different packet flows with each group having distinct latency and bandwidth requirements. For more information on QoS, see the "Configuring QoS for Voice" chapter in this guide, the *Cisco IOS Quality of Service Solutions Configuration Guide,* and the *Cisco IOS Quality of Service Solutions Command Reference*.



Configuring Voice over IP

This chapter provides an overview of Voice over IP (VoIP) technology and gives step-by-step configuration tasks. The chapter contains the following sections:

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" in the "Using Cisco IOS Software" chapter.

Voice over IP Overview

VoIP is a Layer 3 network protocol that uses various Layer 2 point-to-point or link-layer protocols such as PPP, Frame Relay, or ATM for its transport. VoIP enables Cisco routers, access servers, and multiservice access concentrators to carry and send voice and fax traffic over an IP network. In VoIP, digital signal processors (DSPs) segment the voice signal into frames and store them in voice packets. These voice packets are transported via IP in compliance with a voice communications protocol or standard such as H.323, Media Gateway Control Protocol (MGCP), or Session Initiation Protocol (SIP).

Table 6 shows the relationship between the Open System Interconnection (OSI) reference model and the protocols and functions of VoIP network elements.

Table 6 Relationship of OSI Reference Model to VoIP Protocols and Functions

OSI Layer Number	OSI Layer Name	VoIP Protocols and Functions
7	Application	NetMeeting/Applications
6	Presentation	Codecs
5	Session	H.323/MGCP/SIP
4	Transport	RTP/TCP/UDP

Table 6 Relationship of OSI Reference Model to VoIP Protocols and Functions (continued)

OSI Layer Number	OSI Layer Name	VoIP Protocols and Functions
3	Network	IP
2	Data Link	Frame Relay, ATM, Ethernet, PPP, MLP, and more

Cisco IOS software supports the following call control protocols and standards in Release 12.2:

- H.323—the International Telecommunication Union-Telecommunications Standardization Sector (ITU-T) specification for sending voice, video, and data across a network. The H.323 specification includes several related standards, such as H.225 (call control), H.235 (security), H.245 (media path and parameter negotiation), and H.450 (supplementary services). For more information, see the "H.323 Overview" chapter in this configuration guide.
- MGCP—Media Gateway Control Protocol, an Internet Engineering Task Force (IETF) draft standard for controlling voice gateways through IP networks. For more information, see the "Configuring MGCP and Related Protocols" chapter in this configuration guide.
- SIP—Session Initiation Protocol, defined in IETF RFC 2543. For more information, see the "Configuring SIP" chapter in this guide.

VoIP protocols typically use Real-time Transport Protocol (RTP) for the media stream or speech path. RTP uses User Datagram Protocol (UDP) as its transport protocol. Voice signaling traffic often uses Transmission Control Protocol (TCP) as its transport medium. The IP layer provides routing and network-level addressing; the data-link layer protocols control and direct the transmission of the information over the physical medium.

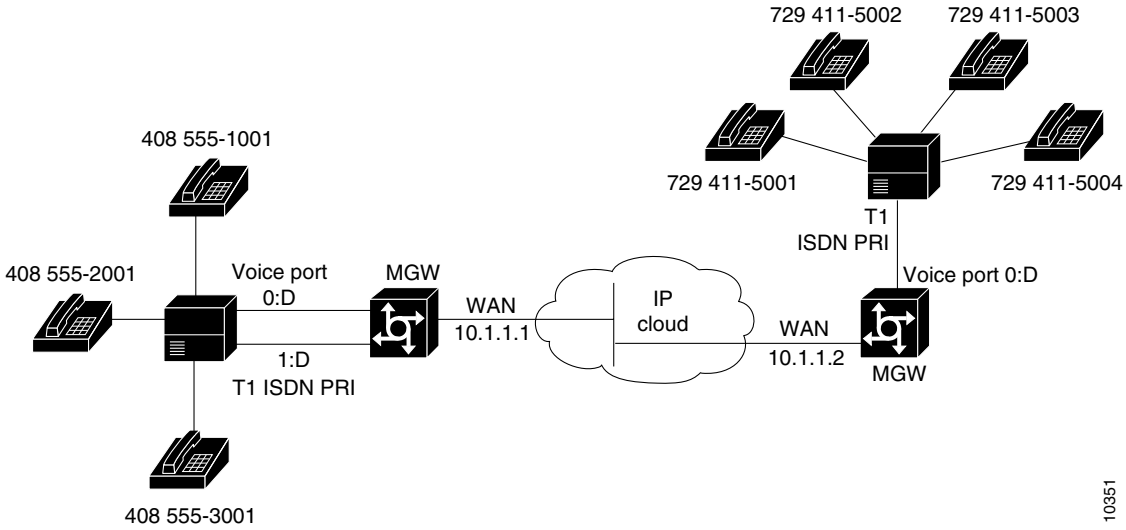
The main factor in choosing between VoIP and the Layer 2 VoFR and VoATM transport alternatives is interworking with other voice or multimedia applications. Generally speaking, Voice over Frame Relay (VoFR) and Voice over ATM (VoATM) are effective WAN transport technologies and are more bandwidth-efficient than VoIP. But VoFR and VoATM cannot be deployed over LANs or to the desktop. VoIP is the predominant form of voice-over-packet deployed today, and, for implementing voice applications, it is usually the only choice even if the first step in network deployment is pure transport between existing PBXs.

VoIP leverages the entire Internet and Intranet IP infrastructure for routing, making it easy to design any-to-any calling in a VoIP network. VoIP also allows multivendor interworking, which is more difficult to achieve with VoFR and VoATM applications because standards for those solutions have only recently emerged.

Cisco VoIP is frequently used in two primary applications:

- To provide a central-site telephony termination facility for voice traffic coming from multiple voice-equipped remote office facilities. Figure 19 illustrates this application using Cisco AS5300 universal access servers as the central-site telephony termination devices.

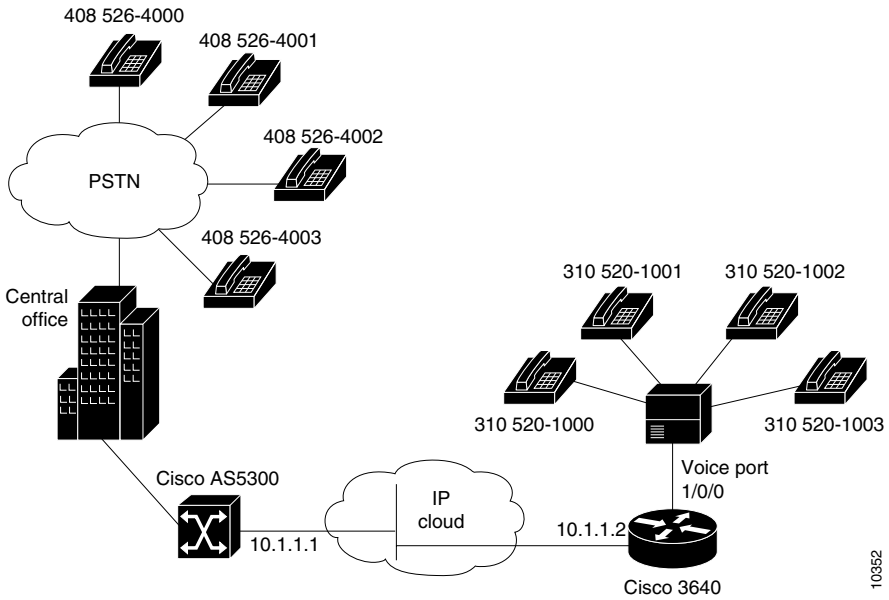
Figure 19 VoIP Used as a Central-Site Telephony Termination Facility



- To provide Public Switched Telephone Network (PSTN) gateway functionality for Internet telephone traffic. Cisco VoIP used in this scenario leverages the standardized use of H.323-based Internet telephone client applications. In the case of a device with extensive capacity running VoIP (such as the Cisco AS5800 universal access server), the functionality provided is equivalent to that of a carrier-class switch.

  Figure 20 illustrates this application, using a Cisco AS5300 as the PSTN gateway.

Figure 20 VoIP Used as a PSTN Gateway for Internet Telephone Traffic



To use VoIP, you must install the appropriate hardware in your Cisco device: for example, a voice-specific port adapter or network module. The specific voice hardware required depends on the router or access server used. The number of ports or channels available for sending VoIP data depends on the capacity of the specific voice hardware installed. For more information about the physical characteristics, capacity, installation, or configuration of voice hardware, refer to the online documentation for your router or access server.

VoIP Benefits

VoIP offers the following benefits:

- Toll bypass (either one- or two-stage toll bypass, depending on the environment in which VoIP is deployed)
- Remote PBX presence over WANs
- PSTN voice-traffic and fax-traffic offload
- Universally accessible voice-mail and fax-mail services
- Unified voice and data trunking
- Plain old telephone service (POTS)-Internet telephony gateways
- Support for Microsoft NetMeeting when a Cisco router is used as a voice gateway

VoIP Call Processing

Before configuring VoIP on a Cisco router or access server, it helps to have a high-level understanding of what happens when you place a VoIP call. The following sequence outlines the general flow of a two-party VoIP voice call using H.323:

1. The caller picks up the handset, signaling an off-hook condition to the signaling application layer of VoIP.
2. The session application layer of VoIP issues a dial tone and waits for the caller to dial.
3. When the caller dials the number, the dialed digits are accumulated and stored by the session application.
4. After enough digits are accumulated to match a configured destination pattern, the telephone number is mapped to an IP host via the dial plan mapper. The IP host has a direct connection to the destination telephone number or a PBX that is responsible for completing the call to the configured destination pattern.
5. The session application runs the H.323 session protocol to establish a transmission and a reception channel for each direction over the IP network. If the call is being handled by a PBX, the PBX forwards the call to the destination telephone. If Resource Reservation Protocol (RSVP) has been configured, the RSVP reservations are put into effect to achieve the desired quality of service (QoS) over the IP network.
6. The coder-decoders (codecs) are enabled for both ends of the connection and the conversation proceeds using RTP/UDP/Internet Protocol (IP) as the protocol stack. Voice signals are digitized, compressed, packaged into discrete packets, and transported over the network.

7. Any call-progress indications or other signals that can be carried in-band are cut through the voice path as soon as the end-to-end audio channel is established. Signaling that can be detected by the voice ports (for example, in-band dual tone multifrequency [DTMF] digits after the call setup is complete) is also trapped by the session application at either end of the connection and carried over the IP network encapsulated in Real Time Conferencing Protocol (RTCP) using the RTCP APP extension mechanism.
8. When either end of the call hangs up, the RSVP reservations are torn down (if RSVP is used) and the session ends. Each end becomes idle, waiting for the next off-hook condition to trigger another call setup.

VoIP Prerequisite Tasks

Before configuring a Cisco router, access server, or gateway to use VoIP, complete the following tasks:

- Establish a working IP network in which delay (as measured by ping tests) and jitter are minimized. For more information about configuring IP, refer to the "IP Overview," "Configuring IP Addressing," and "Configuring IP Services" chapters in the *Cisco IOS IP Configuration Guide*, Release 12.2.
- Install a voice network module (VNM), voice feature card (VFC), or universal port dial feature card into the appropriate slot of your Cisco router, access server, or gateway. For more information about the physical characteristics, capacity, memory requirements, and installation instructions for the hardware you are installing, refer to the appropriate platform-specific hardware documentation.
- Make sure your router, access server, or gateway has sufficient DRAM installed to support VoIP, and make sure you are running a version and image of Cisco IOS software that supports VoIP. For more information, refer to the release notes for the platform you are using and the version of Cisco IOS you are running, or use the Feature Navigator tool on Cisco.com.
- Complete basic configuration of your router, access server, or gateway. For more information about these basic configuration tasks, refer to the "Configuring H.323 Gateways," "Configuring Voice Ports," and "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapters of this configuration guide.
- Formulate the beginning of a dial plan that includes the following:
  - Logical network diagram showing voice ports and components to which they connect, including telephones, fax machines, PBX or key systems, other voice devices that require connection, and voice-enabled routers.
  - Connection details, including physical interfaces, relevant LAN and WAN ports, and all voice ports; for each WAN, the type (Frame Relay, PPP, etc.); for Frame Relay, relevant PVCs and link access rates.
  - Phone numbers or extensions for each voice port, logically laid out and consistent with existing private dial plans and external dialing schemes.
- Establish a working telephony network based on your company dial plan.
- Integrate your dial plan and telephony network into your existing IP network topology. In general, we recommend the following practices:
  - Make routing or dialing transparent to users; for example, avoid secondary dial tones from secondary switches, where possible.
  - Contact your PBX vendor for instructions about how to reconfigure the appropriate PBX interfaces.

VoIP Network Design Considerations

You must have a well-engineered network end-to-end when running delay-sensitive applications such as VoIP. Fine-tuning your network to adequately support VoIP involves a series of protocols and features geared toward improving quality of service (QoS).

Quality of service refers to the ability of a network to provide differentiated service to selected network traffic over various underlying technologies. QoS is not inherent in a network infrastructure. Rather, you institute QoS by strategically enabling appropriate QoS features throughout your network.

Cisco IOS software provides many tools for enabling QoS on your backbone, such as Random Early Detection (RED), Weighted Random Early Detection (WRED), fancy queueing (meaning custom, priority, or weighted fair queueing), IP RTP priority, low-latency queueing (LLQ), and IP precedence. To configure your IP network for real-time voice traffic, you must take into consideration the entire scope of your network and then select the appropriate QoS tool or tools. For complete information about any of these topics, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.2. In addition, refer to the "Configuring QoS for Voice" chapter in this configuration guide.

Remember that to improve voice network performance, QoS must be configured throughout your network, not just on the Cisco devices running VoIP. Not all QoS techniques are appropriate for all network routers. Edge routers and backbone routers in your network do not necessarily perform the same operations; the QoS tasks they perform might differ as well. To configure your IP network for real-time voice traffic, you must consider the functions of both edge and backbone routers in your network and then select the appropriate QoS tool or tools.

VoIP Quality of Service Tips

This section explains the quality issues that you should consider when building VoIP networks and offers a few tips about configuring VoIP with the appropriate QoS. For detailed information on these topics, refer to "Voice Quality Tuning Commands" in the "Configuring Voice Ports" chapter.

Voice traffic differs from data traffic in the following ways:

- Data is often bursty by nature; voice is deterministic (smooth).
- Data applications resend dropped packets; voice applications can only conceal dropped packets.
- Data applications can usually tolerate some delay; voice applications must minimize delay, so that the recipient does not hear clips in the transmission.

These differences mandate the use of QoS strategies to give strict priority to voice traffic, ensuring reliable delivery and minimal delay for networks that carry both voice and data.

Delay

Delay is the time it takes for VoIP packets to travel between two endpoints. Because of the speed of network links and the processing power of intermediate devices, some delay is expected; however, you should attempt to minimize this delay.

The human ear normally accepts a delay of about 150 milliseconds (ms) without noticing problems. (The ITU G.114 standard recommends no more than 150 ms of one-way delay.) When delay exceeds 150 ms, a conversation becomes more and more like a citizens band (CB) radio interchange in which one person must wait for the other to stop speaking before beginning to talk. This type of delay is often evident on international long-distance calls. You can measure delay fairly easily by using ping tests at various times of the day with different network traffic loads. If network delay is excessive, reduce it before deploying VoIP in your network.

Jitter

Although delay can cause unnatural starting and stopping of conversations, variable-length delays (also known as *jitter*) can cause a conversation to break and become unintelligible. Jitter is not usually a problem with PSTN calls because the bandwidth of calls is fixed. However, in VoIP networks in which existing data traffic might be bursty, jitter can become a problem. Cisco voice gateways have built-in dejitter buffering to compensate for a certain amount of jitter, but if jitter is constant on a network, identify the source and control it before deploying a VoIP network.

Serialization

Serialization is a term that describes what happens when a router attempts to send both voice and data packets through an interface. In general, voice packets are very small (80 to 256 bytes), and data packets can be very large (1500 to 18,000 bytes). On relatively slow links, such as WAN connections, large data packets can take a long time to send onto the wire. When these large packets are mixed with smaller voice packets, the excessive transmission time can lead to both delay and jitter. You can use fragmentation to reduce the size of the data packets so that the delay and jitter also decrease.

Bandwidth Consumption

Traditional voice conversations consume 64 kbps of network bandwidth. When this voice traffic is run though a VoIP network, it can be compressed and digitized by digital signal processors (DSPs built into the routers. This compression can reduce the calls to sizes as small as 5.3 kbps for voice samples. After the packets go onto the IP network, the appropriate IP/UDP/RTP headers must be added. This can add a substantial amount of bandwidth to each call (about 40 bytes per packet). Technologies such as RTP header compression, however, can reduce the IP header overhead to about two bytes. In addition, VAD does not send any packets unless there is active speech.

VoIP Configuration Task List

To configure VoIP on a Cisco router or access server, complete the following tasks:

Step 1 Configure your IP network for real-time voice traffic. Fine-tuning your network to adequately support VoIP involves a series of protocols and features designed to improve QoS. To configure your IP network for real-time voice traffic, consider the entire scope of your network. Then select and configure the appropriate QoS tool or tools.

Refer to "Configuring VoIP over Frame Relay" section on page 53, and the "Configuring QoS for Voice" chapter for information about how to select and configure the appropriate QoS tools to optimize voice traffic on your network.

Step 2 If you plan to run VoIP over Frame Relay, you must take certain factors into consideration when configuring VoIP for it to run smoothly over Frame Relay. For example, a public Frame Relay cloud provides no guarantees for QoS. Refer to the "Configuring VoIP over Frame Relay" section on page 53 for information about deploying VoIP over Frame Relay.

Step 3 Configure dial peers. Use the **dial-peer voice** command to define dial peers and switch to the dial-peer configuration mode. Each dial peer defines the characteristics associated with a call leg. A call leg is a discrete segment of a call connection that lies between two points in the connection. An end-to-end call consists of four call legs, two from the perspective of the source access server, and two from the perspective of the destination access server. Dial peers are used to apply attributes to call legs and to identify call origin and destination. There are two types of dial peers used for VoIP:

- POTS—Dial peer describing the characteristics of a traditional telephony network connection. POTS dial peers point to a particular voice port on a voice network device. To configure a POTS dial peer, you must configure the associated telephone number and the logical interface. Use the **destination-pattern** command to associate a telephone number with a POTS peer. Use the **port** command to associate a specific logical interface with a POTS peer. In addition, you can specify direct inward dialing for a POTS peer by using the **direct-inward-dial** command.
- VoIP—Dial peer describing the characteristics of the IP network connection. VoIP dial peers point to specific VoIP devices. To configure a VoIP dial peer, you must configure the associated destination telephone number and a destination IP address. Use the **destination-pattern** command to define the destination telephone number associated with a VoIP peer. Use the **session target** command to specify a destination IP address for a VoIP peer.

  Refer to the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide for additional information about dial-peer characteristics and configuring dial peers.

Step 4 Configure number expansion. Use the **num-exp** command to configure number expansion if your telephone network is configured so that you can reach a destination by dialing only a portion (an extension number) of the full E.164 telephone number. Refer to the "Configuring Digit Manipulation Features" section of the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter of this guide for information about number expansion.

Step 5 Optimize dial peer and network interface configurations. You can use VoIP dial peers to define characteristics such as codec, voice activity detection (VAD), and additional QoS parameters (when RSVP is configured). If you have configured RSVP, use either the **req-qos** or **acc-qos** command to configure QoS parameters. Use the **codec** command to configure specific voice coder rates. Use the **vad** command to disable voice activation detection and the transmission of silence packets. Refer to the "Configuring Dial Plan Options for VoIP Dial Peers" section of the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this guide for additional information about optimizing dial-peer characteristics.

Step 6 Configure voice ports. In general, voice-port commands define the characteristics associated with a particular voice-port signaling type. The following voice signaling types are supported:

- FXO—Foreign Exchange Office interface
- FXS—The Foreign Exchange Station interface
- E&M—The "ear and mouth" interface (also called the "earth and magnet interface, or the "recEive and transMit" interface)

Under most circumstances, the default voice-port command values are adequate to configure FXO and FXS ports to transport voice data over your existing IP network. Because of the inherent complexities involved with PBX networks, E&M ports might need specific voice-port values configured, depending on the specifications of the devices in your telephony network. For information about configuring voice ports, refer to the "Configuring Voice Ports" chapter in this guide.

Configuring VoIP over Frame Relay

You must consider certain factors when configuring VoIP to ensure that it runs smoothly over Frame Relay. A public Frame Relay cloud provides no guarantees for QoS. For real-time traffic to be sent in a timely manner, the data rate must not exceed the committed information rate (CIR) or packets may be dropped. In addition, Frame Relay traffic shaping and RSVP are mutually exclusive. Remembering this is particularly important if multiple data link connection identifiers (DLCIs) are carried on a single interface.

For Frame Relay links with slow output rates (less than or equal to 64 kbps) in which data and voice are being sent over the same permanent virtual circuit (PVC), we recommend the following solutions:

- Separate DLCIs for voice and data—By providing a separate subinterface for voice and data, you can use the appropriate QoS tool for each line. For example, with each DLCI using 32 kbps of a 64-kbps line, you could do the following:
  - Apply adaptive traffic shaping to both DLCIs.
  - Use RSVP or IP Precedence to prioritize voice traffic.
  - Use compressed RTP to minimize voice packet size.
  - Use weighted fair queueing to manage voice traffic.
- Lower the maximum transmission unit (MTU) size—Voice packets are generally small. With a lower MTU size (for example, 300 bytes), large data packets can be broken up into smaller data packets that can more easily be interwoven with voice packets.

Note Some applications do not support a smaller MTU size. If you decide to lower the MTU size, use the **ip mtu** command; this command affects only IP traffic.

Note Lowering the MTU size affects data throughput speed.

- CIR equal to line rate—Make sure that the data rate does not exceed the CIR. One way you can make sure that the data rate does not exceed the CIR is through generic traffic shaping. For example, you could do the following:
  - Use IP precedence to prioritize voice traffic.
  - Use compressed RTP to minimize voice packet header size.
- Traffic shaping—Use adaptive traffic shaping to throttle back the output rate based on the backward explicit congestion notification (BECN). If the feedback from the switch is ignored, both data and voice packets might be discarded. Because the Frame Relay switch does not distinguish between voice and data packets, voice packets could be discarded, resulting in a deterioration of voice quality. For example, you could do the following:
  - Use compressed RTP, reduced MTU size, and adaptive traffic shaping based on BECN to hold the data rate to the CIR.
  - Use generic traffic shaping to obtain a low interpacket wait time. For example, set Bc to 4000 to obtain an interpacket wait of 125 ms.

Note We recommend FRF.12 fragmentation setup rules for VoIP connections over Frame Relay. For more information, refer to the "Configuring Voice over Frame Relay" chapter.

VoIP Configuration Examples

This section contains the following configuration examples:

VoIP over Frame Relay Configuration Example

For Frame Relay, it is customary to configure a main interface and one subinterface per permanent virtual circuit (PVC). The following example configures a Frame Relay main interface and a subinterface so that voice and data traffic can be successfully transported:

```
interface Serial0/0
 ip mtu 300
 no ip address
 encapsulation frame-relay
 no ip route-cache
 no ip mroute-cache
 fair-queue 64 256 1000
 frame-relay ip rtp header-compression

interface Serial0/0.1 point-to-point
 ip mtu 300
 ip address 40.0.0.7 255.0.0.0
 no ip route-cache
 no ip mroute-cache
 bandwidth 64
 traffic-shape rate 32000 4000 4000
 frame-relay interface-dlci 16
 frame-relay ip rtp header-compression
```

In this configuration example, the main interface has been configured as follows:

- Maximum Transmission Unit (MTU) size of IP packets is 300 bytes.
- No IP address is associated with this serial interface. The IP address must be assigned for the subinterface.
- Encapsulation method is Frame Relay.
- Fair queueing is enabled.
- IP RTP header compression is enabled.

The subinterface has been configured as follows:

- MTU size is inherited from the main interface.
- IP address for the subinterface is specified.
- Bandwidth is set to 64 kbps.
- Generic traffic shaping is enabled with 32 kbps CIR where Bc = 4000 bits and Be = 4000 bits.
- Frame Relay DLCI number is specified.
- IP RTP header compression is enabled.

Note When traffic bursts over the CIR, the output rate is held at the speed configured for the CIR (for example, traffic will not go beyond 32 kbps if CIR is set to 32 kbps).

For more information about Frame Relay, refer to the "Configuring Frame Relay" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide*.

VoIP for the Cisco 3600 Series Configuration Examples

The actual VoIP configuration procedure you complete depends on the topology of your voice network. The following configuration examples are a starting point. Of course, these configuration examples must be customized to reflect your network topology.

Configuration examples are supplied for the following sections:

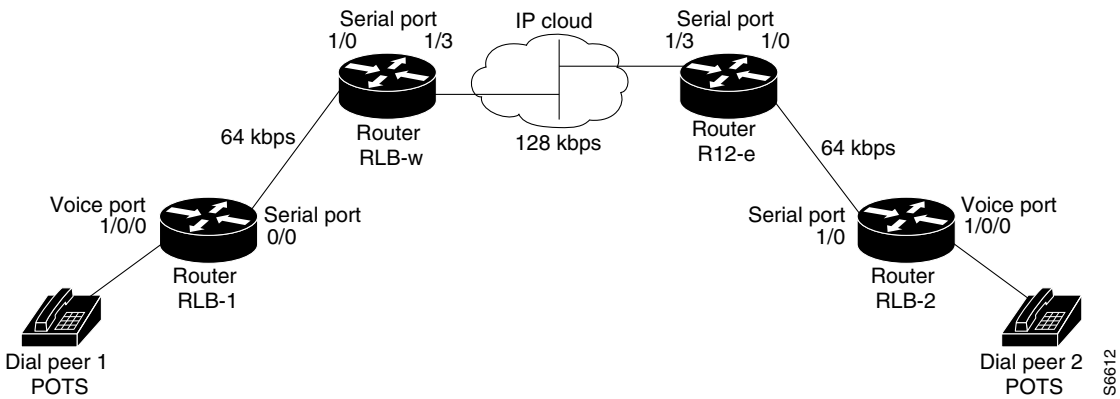
FXS-to-FXS Connection Using RSVP

The following example shows how to configure VoIP for simple FXS-to-FXS connections.

In this example, a very small company of two offices has decided to integrate VoIP into its existing IP network. One basic telephony device is connected to Router RLB-1; therefore Router RLB-1 is configured for one POTS dial peer and one VoIP dial peer. Router RLB-w and Router R12-e establish the WAN connection between the two offices. Because one POTS telephony device is connected to Router RLB-2, it is also configured for only one POTS peer and one VoIP peer.

Note In this example, only the calling end (Router RLB-1) is requesting RSVP. Figure 21 illustrates the topology of this FXS-to-FXS connection example.

Figure 21 FXS-to-FXS Connection Example



Router RLB-1

```
hostname rlb-1

! Create voip dial peer 10
dial-peer voice 10 voip

! Define its associated telephone number and IP address
 destination-pattern +4155554000
 session target ipv4:40.0.0.1

! Request RSVP
 req-qos guaranteed-delay

! Create pots dial peer 1
dial-peer voice 1 pots

! Define its associated telephone number and voice port
 destination-pattern +4085554000
 port 1/0/0

! Configure serial interface 0/0
interface Serial0/0
 ip address 10.0.0.1 255.0.0.0
 no ip mroute-cache

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 48 48
 fair-queue 64 256 36
 clockrate 64000

router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

Router RLB-w

```
hostname rlb-w

! Configure serial interface 1/0
interface Serial1/0
 ip address 10.0.0.2 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3

! Configure serial interface 1/3
interface Serial1/3
 ip address 20.0.0.1 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25
```

```
! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3

! Configure IGRP
router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

Router R12-e

```
hostname r12-e

! Configure serial interface 1/0
interface Serial1/0
 ip address 40.0.0.2 25.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3

! Configure serial interface 1/3
interface Serial1/3
 ip address 20.0.0.2 255.0.0.0

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3
 clockrate 128000

! Configure IGRP
router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

Router RLB-2

```
hostname r1b-2

! Create pots dial peer 2
dial-peer voice 2 pots

! Define its associated telephone number and voice port
 destination-pattern +4155554000
 port 1/0/0

! Create voip dial peer 20
dial-peer voice 20 voip
!Define its associated telephone number and IP address
 destination-pattern +4085554000
 session target ipv4:10.0.0.1

! Configure serial interface 0/0
interface Serial0/0
```

```
 ip address 40.0.0.1 255.0.0.0
 no ip mroute-cache

! Configure RTP header compression
 ip rtp header-compression
 ip rtp compression-connections 25

! Enable RSVP on this interface
 ip rsvp bandwidth 96 96
 fair-queue 64 256 3
 clockrate 64000

! Configure IGRP
router igrp 888
 network 10.0.0.0
 network 20.0.0.0
 network 40.0.0.0
```

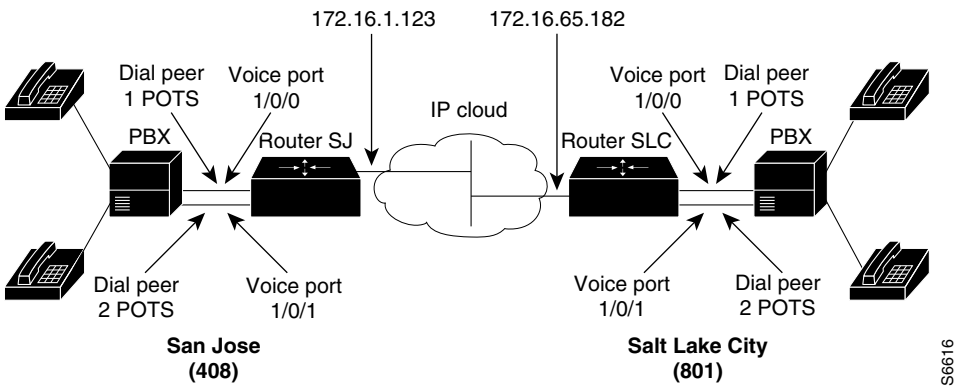
Linking PBX Users with E&M Trunk Lines

The following example shows how to configure VoIP to link PBX users with E&M trunk lines.

In this example, a company wants to connect two offices: one in San Jose, California, and the other in Salt Lake City, Utah. Each office has an internal telephone network using a PBX that is connected to the voice network by an E&M interface. Both the Salt Lake City and the San Jose offices are using E&M Port Type II with 4-wire operation and Immediate Start signaling. Each E&M interface connects to the router using two voice interface connections. Users in San Jose dial 8569 and then the extension number to reach a destination in Salt Lake City. Users in Salt Lake City dial 4527 and then the extension number to reach a destination in San Jose.

Figure 22 illustrates the topology of this connection example.

Figure 22 Linking PBX Users with E&M Trunk Lines Example



Note This example assumes that the company already has working IP connection between its two remote offices.

Router SJ

```
hostname sanjose

!Configure pots dial peer 1
dial-peer voice 1 pots
```

```
 destination-pattern 555....
 port 1/0/0

!Configure pots dial peer 2
dial-peer voice 2 pots
 destination-pattern 555....
 port 1/0/1

!Configure voip dial peer 3
dial-peer voice 3 voip
 destination-pattern 119....
 session target ipv4:172.16.65.182

!Configure the E&M interface
voice-port 1/0/0
 signal immediate
 operation 4-wire
 type 2

voice-port 1/0/1
 signal immediate
 operation 4-wire
 type 2

!Configure the serial interface
interface serial 0/0
 description serial interface type dce (provides clock)
 clock rate 2000000
 ip address 172.16.1.123
 no shutdown
```

Router SLC

```
hostname saltlake

!Configure pots dial peer 1
dial-peer voice 1 pots
 destination-pattern 119....
 port 1/0/0

!Configure pots dial peer 2
dial-peer voice 2 pots
 destination-pattern 119....
 port 1/0/1

!Configure voip dial peer 3
dial-peer voice 3 voip
 destination-pattern 555....
 session target ipv4:172.16.1.123

!Configure the E&M interface
voice-port 1/0/0
 signal immediate
 operation 4-wire
 type 2

voice-port 1/0/0
 signal immediate
 operation 4-wire
 type 2

!Configure the serial interface
interface serial 0/0
 description serial interface type dte
```

```
 ip address 172.16.65.182
 no shutdown
```

Note PBXs should be configured to pass all DTMF signals to the router. We recommend that you do not configure store and forward tone.

Note If you change the gain or the telephony port, make sure that the telephony port still accepts DTMF signals.

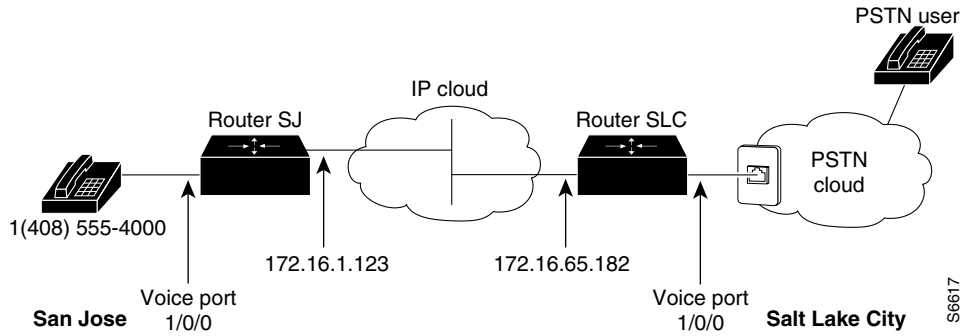
PSTN Gateway Access Using FXO Connection

The following example shows how to configure VoIP to link users with the PSTN gateway using an FXO connection.

In this example, users connected to Router SJ in San Jose, California, can reach PSTN users in Salt Lake City, Utah, via Router SLC. Router SLC in Salt Lake City is connected directly to the PSTN through an FXO interface.

Figure 23 illustrates the topology of this connection example.

Figure 23 PSTN Gateway Access Using FXO Connection Example



Note This example assumes that the company already has a working IP connection between its two remote offices.

Router SJ

```
! Configure pots dial peer 1
dial-peer voice 1 pots
 destination-pattern +14085554000
 port 1/0/0

! Configure voip dial peer 2
dial-peer voice 2 voip
 destination-pattern 9...........
 session target ipv4:172.16.65.182

! Configure the serial interface
interface serial 0/0
 clock rate 2000000
```

```
 ip address 172.16.1.123
 no shutdown
```

Router SLC

```
! Configure pots dial peer 1
dial-peer voice 1 pots
 destination-pattern 9...........
 port 1/0/0

! Configure voip dial peer 2
dial-peer voice 2 voip
 destination-pattern +14085554000
 session target ipv4:172.16.1.123

! Configure serial interface
interface serial 0/0
 ip address 172.16.65.182
 no shutdown
```

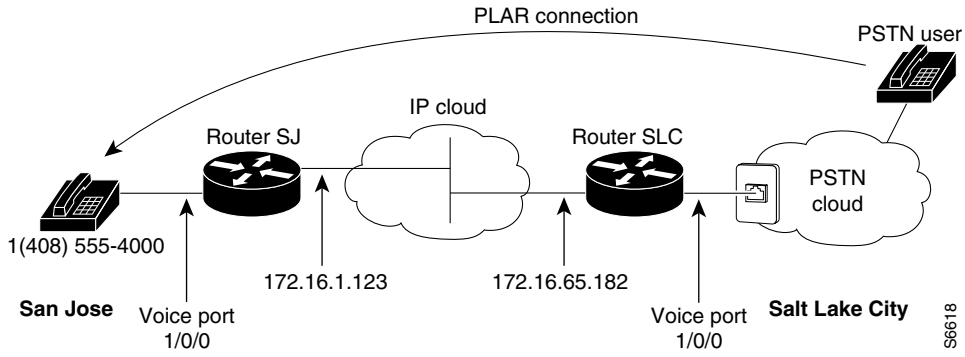
PSTN Gateway Access Using FXO Connection (PLAR Mode)

The following example shows how to configure VoIP to link users with the PSTN gateway using an FXO connection in private line auto-ringdown (PLAR) mode.

In this example, PSTN users in Salt Lake City, Utah, can dial a local number and establish a private-line connection in a remote location. As in the preceding example, Router SLC in Salt Lake City is connected directly to the PSTN through an FXO interface.

Figure 24 illustrates the topology of this connection example.

Figure 24 PSTN Gateway Access Using FXO Connection (PLAR Mode)



Note This example assumes that the company already has a working IP connection between its two remote offices.

Router SJ

```
! Configure pots dial peer 1
dial-peer voice 1 pots
 destination-pattern +14085554000
 port 1/0/0
```

```
! Configure voip dial peer 2
dial-peer voice 2 voip
 destination-pattern 9...........
 session target ipv4:172.16.65.182

! Configure the serial interface
interface serial 0/0
 clock rate 2000000
 ip address 172.16.1.123
 no shutdown
```

Router SLC

```
! Configure pots dial peer 1
dial-peer voice 1 pots
 destination-pattern 9...........
 port 1/0/0

! Configure voip dial peer 2
dial-peer voice 2 voip
 destination-pattern +14085554000
 session target ipv4:172.16.1.123

! Configure the voice-port
voice-port 1/0/0
connection plar 14085554000

! Configure the serial interface
interface serial 0/0
 ip address 172.16.65.182
 no shutdown
```

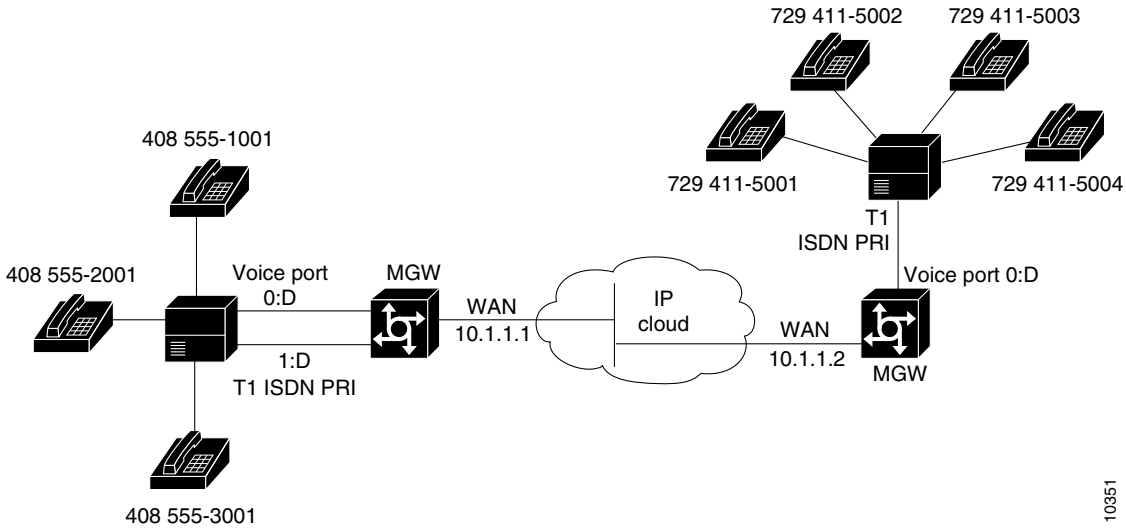
VoIP for the Cisco AS5300 Configuration Example

This configuration example should give you a starting point in your configuration process. The actual VoIP configuration procedure you complete depends on the topology of your voice network. These configuration examples must be customized to reflect your network topology.

Linking PBX Users to a T1 ISDN PRI Interface

This example describes how to configure VoIP to link PBX users with T1 channels configured for ISDN PRI signaling. In this example, the company has already established a working IP connection between its two remote offices, one in San Jose, California, and the other in Research Triangle Park (RTP), North Carolina. Figure 25 illustrates the topology of this example.

Figure 25 Linking PBX Users to a T1 ISDN PRI Interface Example



Each office has an internal telephone network using a PBX that is connected to the voice network by T1 interfaces. The San Jose office, located to the left of the IP cloud, has two T1 connections; the RTP office, located to the right of the IP cloud, has only one. Both offices are using PRI signaling for the T1 connections.

To reach a destination in RTP, callers in San Jose pick up the handset, hear a primary dial tone, and dial 9, 411, and the destination extension number. To reach a destination in San Jose, callers in RTP pick up the handset, hear a primary dial tone, and dial 4. After dialing 4, callers hear a secondary dial tone. They then dial 555 and the extension number.

Configuration for San Jose Access Server

The first part of this configuration example defines dial-in access, including configuring the T1 lines and the ISDN D-channel parameters:

```
hostname sanjose
!
! Define the telephone company's switch type
isdn switch-type primary-5ess
!
! Configure T1 PRI for line 1
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
! Configure T1 PRI for line 2
controller T1 1
 framing esf
 clock source line secondary
 linecode b8zs
 pri-group timeslots 1-24
!
! Configure the ISDN D channel for each ISDN PRI line
! Serial interface 0:23 is the D channel for controller T1 0
!
interface Serial0:23
 isdn incoming-voice modem
!
```

```
! Serial interface 1:23 is the D channel for controller T1 1
interface Serial1:23
 isdn incoming-voice modem
```

The next part of this example configures number expansion:

```
! Configure number expansion.
num-exp 555.... 1408555....
num-exp 4115... 17294115...
```

The next part of this example configures the POTS and VoIP dial peers:

```
! Configure POTS dial peer 1 using the first T1
dial-peer voice 1 pots
 prefix 6
 dest-pat 1408555....
 port 0:D
!
! Configure POTS dial-peer 2 using the first T1
dial-peer voice 2 pots
 prefix 7
 dest-pat 1408555....
 port 0:D
!
! Configure POTS dial-peer 3 using the second T1
dial-peer voice 3 pots
 prefix 5
 dest-pat 1408555....
 port 1:D
!
! Configure VoIP dial-peer 4
dial-peer voice 4 voip
 dest-pat 17294115...
 session-target ipv4:10.1.1.2
```

Configuration for RTP Access Server

The first part of this configuration example defines dial-in access, including configuring the T1 line and the ISDN D-channel parameters:

```
hostname rtp

! Define the telephone company's switch type
isdn switch-type primary-5ess

! Configure T1 PRI for line 1
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
! Configure the ISDN D channel for ISDN PRI line 1
! Serial interface 0:23 is the D channel for controller T1 0
interface Serial0:23
 ip address 7.1.1.10 255.255.255.0
 encapsulation ppp
 isdn incoming-voice modem
 dialer-group 1
 ppp authentication chap
```

The next part of this example configures number expansion:

```
! Configure number expansion.
num-exp 555.... 1408555....
num-exp 4115... 17294115...
```

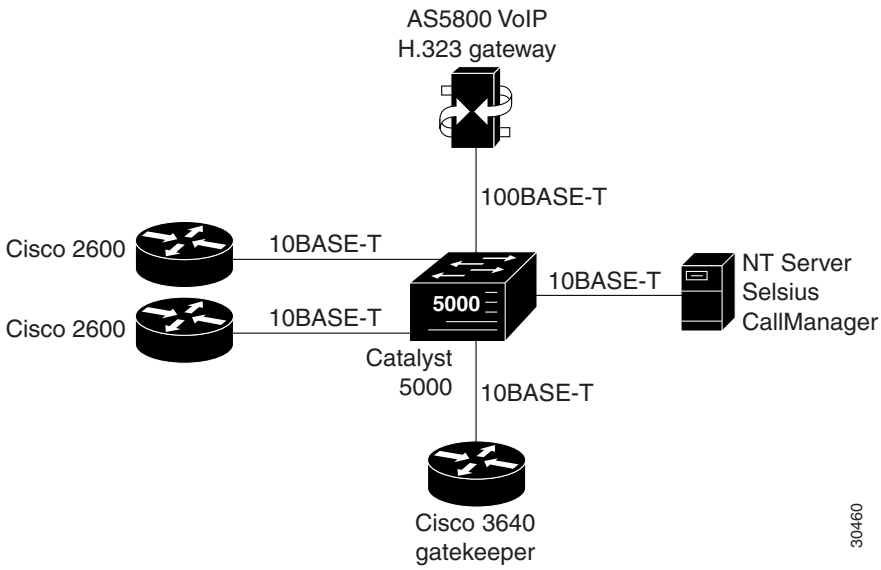
The next part of this configuration example defines the POTS and VoIP peers:

```
! Configure POTS dial-peer 1
dial-peer voice 1 pots
 dest-pat 17294115...
 port 0:D
!
! Configure VoIP dial-peer 5
dial-peer voice 4 voip
 dest-pat 1408555....
 session-target ipv4:10.1.1.1
```

VoIP for the Cisco AS5800 Configuration Example

The following configuration example shows an abbreviated configuration using a Cisco 2600 router and a Cisco AS5800 universal access server as gateways and a Cisco 3600 router as a gatekeeper. Figure 26 shows the network diagram for this particular scenario.

Figure 26 Cisco AS5800 Universal Access Server Acting As a Gateway



Configuring the Cisco 3640 As a Gatekeeper

The following example shows how to configure a Cisco 3640 router as a gatekeeper:

```
! Configure the Ethernet interface to be used at the gatekeeper interface.
interface Ethernet0/1
  ip address 172.30.00.00 255.255.255.0
  no ip directed-broadcast
  no logging event link-status
  no keepalive
```

```
!
! Configure the gatekeeper interface and enable the interface.
gatekeeper
  zone local gk3.gg-dn1 gg-dn1 173.50.00.00
  zone prefix gk3.gg-dn1 21*
  gw-type-prefix 9#* gw ipaddr 173.60.0.0 1720
  gw-type-prefix 6#* gw ipaddr 173.60.0.199 1720
  no use-proxy gk3.gg-dn1 default inbound-to terminal
  no shutdown
!
```

Configuring the Cisco 2600 As a Gateway

The following example shows how to configure a Cisco 2600 series router as a gateway:

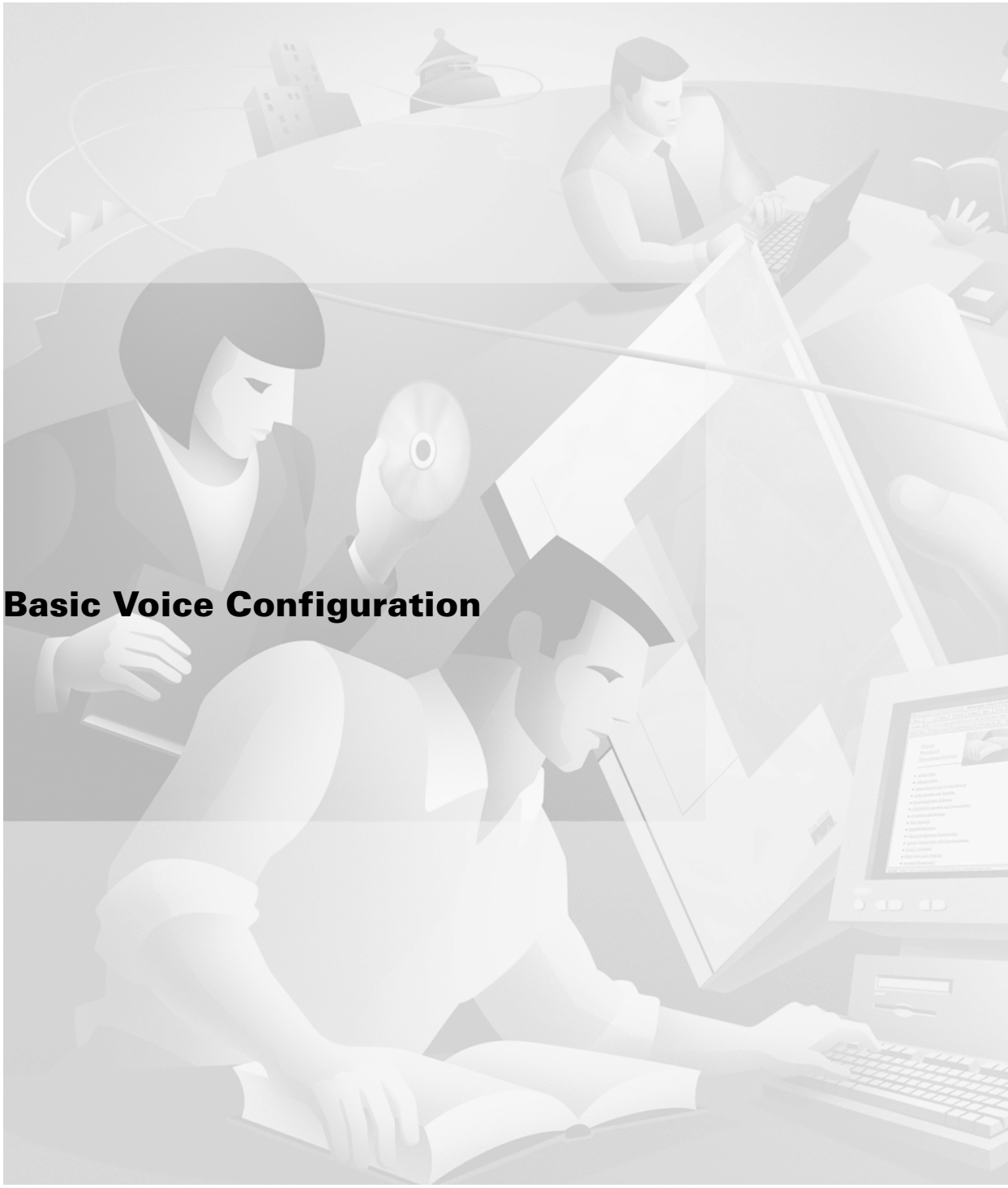
```
! Configure POTS and VoIP dial peers.
dial-peer voice 88 voip
  destination-pattern 11111
  tech-prefix 9#
  session ras
!
dial-peer voice 11 pots
  incoming called-number 11111
  destination-pattern 6#12345
port 1/1/1
prefix 12345
!
! Configure the gateway interface.
interface Ethernet0/0
ip address 173.60.0.199 255.255.255.0
no ip directed-broadcast
no ip mroute-cache
no logging event link-status
no keepalive
no cdp enabled
h323-gateway voip interface
h323-gateway voip id gk3.gg-dn1 ipaddr 173.30.0.0 1719
h323-gateway voip h323-id gw6@gg-dn1
h323-gateway voip tech-prefix 6#
!
```

Configuring the Cisco AS5800 as a Gateway

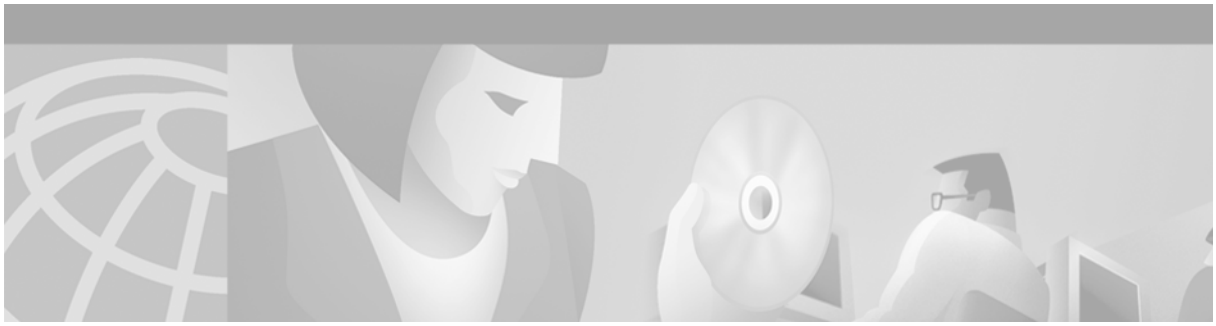
The following example shows how to configure the Cisco AS5800 universal access server as a gateway:

```
! Configure the T1 controller. (This configuration is for a T3 card.)
controller T1 1/0/0:1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
! Configure POTS and VoIP dial peers.
dial-peer voice 11111 pots
 incoming called-number 12345
 destination-pattern 9#11111
 direct-inward-dial
 port 1/0/0:1:D
 prefix 11111
!
dial-peer voice 12345 voip
 destination-pattern 12345
 tech-prefix 6#
```

```
 session target ras
!
! Enable gateway functionality.
gateway
!
! Enable Cisco Express Forwarding.
ip cef
!
! Configure and enable the gateway interface.
interface FastEthernet0/3/0
 ip address 173.60.0.0.255.255.255.0
 no ip directed-broadcast
 no keepalive
 full-duplex
 no cdp enable
 h323-gateway voip interface
 h323-gateway voip id gk3.gg-dn1 ipaddr 173.30.0.0 1719
 h323-gateway voip h323-id gw3@gg-dn1
 h323-gateway voip tech-prefix 9#
!
! Configure the serial interface.(This configuration is for a T3 serial interface.)
interface Serial1/0/0:1:23
 no ip address
 no ip directed-broadcast
 ip mroute-cache
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no cdp enable
```



Basic Voice Configuration



Configuring Voice Ports

Voice ports are found at the intersections of packet-based networks and traditional telephony networks, and they facilitate the passing of voice and call signals between the two networks. Physically, voice ports connect a router or access server to a line from a circuit-switched telephony device in a PBX or the public switched telephone network (PSTN).

Basic software configuration for voice ports describes the type of connection being made and the type of signaling to take place over this connection. Additional commands provide fine-tuning for voice quality, enable special features, and specify parameters to match those of proprietary PBXs.

This chapter includes the following sections:

Not all voice-port commands are covered in this chapter. Some are described in the "Configuring Trunk Connections and Conditioning Features" chapter or the "Configuring ISDN Interfaces for Voice" chapter in this configuration guide. The voice-port configuration commands included in this chapter are fully documented in the *Cisco IOS Voice, Video, and Fax Command Reference*.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Voice Port Configuration Overview

Voice ports on routers and access servers emulate physical telephony switch connections so that voice calls and their associated signaling can be transferred intact between a packet network and a circuit-switched network or device.

For a voice call to occur, certain information must be passed between the telephony devices at either end of the call, such as the devices' on-hook status, the line's availability, and whether an incoming call is trying to reach a device. This information is referred to as signaling, and to process it properly, the devices at both ends of the call segment (that is, those directly connected to each other) must use the same type of signaling.

The devices in the packet network must be configured to convey signaling information in a way that the circuit-switched network can understand. They must also be able to understand signaling information received from the circuit-switched network. This is accomplished by installing appropriate voice hardware in the router or access server and by configuring the voice ports that connect to telephony devices or the circuit-switched network.

The illustrations below show examples of voice port usage.

- In Figure 27, one voice port connects a telephone to the wide-area network (WAN) through the router.
- In Figure 28, one voice port connects to the PSTN and another to a telephone; the router acts like a small PBX.
- Figure 29 shows how two PBXs can be connected over a WAN to provide toll bypass.

Figure 27 Telephone to WAN

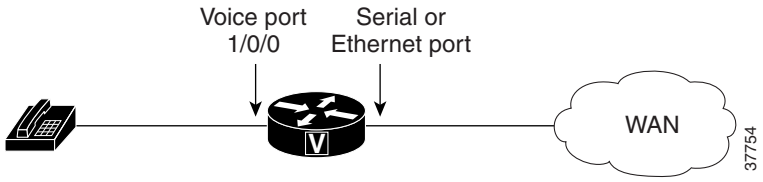


Figure 28 Telephone to PSTN

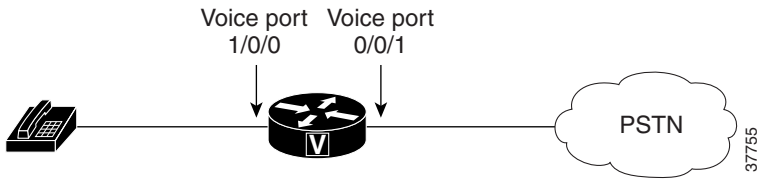


Figure 29 PBX-to-PBX over a WAN

Cisco provides a variety of Cisco IOS commands for flexibility in programming voice ports to match the physical attributes of the voice connections that are being made. Some of these connections are made using analog means of transmission, while others use digital transmission. Table 7 shows the analog and digital voice-port connection support of the router platforms discussed in this chapter.

Table 7 Analog and Digital Voice-port Support on Cisco Routers and Access Servers

Platform	Analog	Digital
Cisco 803 and 804	Yes	No
Cisco 1750	Yes	No
Cisco 2600 series	Yes	Yes
Cisco 3600 series	Yes	Yes
Cisco MC3810	Yes	Yes
Cisco AS5300	No	Yes
Cisco AS5800	No	Yes
Cisco 7200 series	No	Yes
Cisco 7500 series	No	Yes

Telephony Signaling Interfaces

Voice ports on routers and access servers physically connect the router or access server to telephony devices such as telephones, fax machines, PBXs, and PSTN central office (CO) switches. These devices may use any of several types of signaling interfaces to generate information about on-hook status, ringing, and line seizure.

The router's voice-port hardware and software need to be configured to transmit and receive the same type of signaling being used by the device with which they are interfacing so that calls can be exchanged smoothly between the packet network and the circuit-switched network.

The signaling interfaces discussed in this chapter include foreign exchange office (FXO), foreign exchange station (FXS), and receive and transmit (E&M), which are types of analog interfaces. Some digital connections emulate FXO, FXS, and E&M interfaces, and they are discussed in the second half of this chapter. It is important to know which signaling method the telephony side of the connection is using, and to match the router configuration and voice interface hardware to that signaling method.

The next three illustrations show how the different signaling interfaces are associated with different uses of voice ports. In Figure 30, FXS signaling is used for end-user telephony equipment, such as a telephone or fax machine. Figure 31 shows an FXS connection to a telephone and an FXO connection to the PSTN at the far side of a WAN; this might be a telephone at a local office going over a WAN to a router at headquarters that connects to the PSTN. In Figure 32, two PBXs are connected across a WAN by E&M interfaces. This illustrates the path over a WAN between two geographically separated offices in the same company.

Figure 30 FXS Signaling Interfaces

Figure 31 FXS and FXO Signaling Interfaces

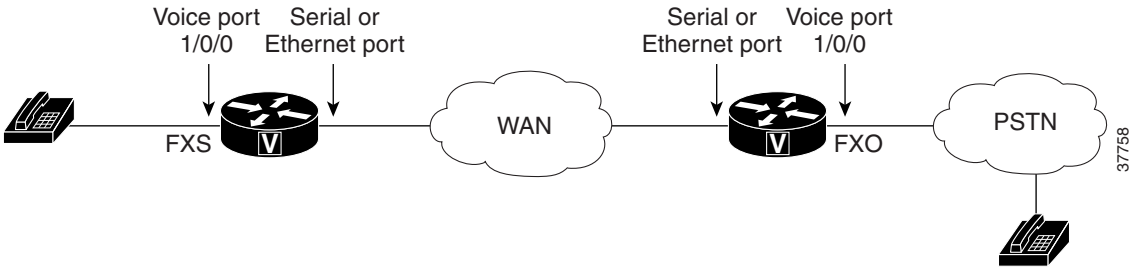
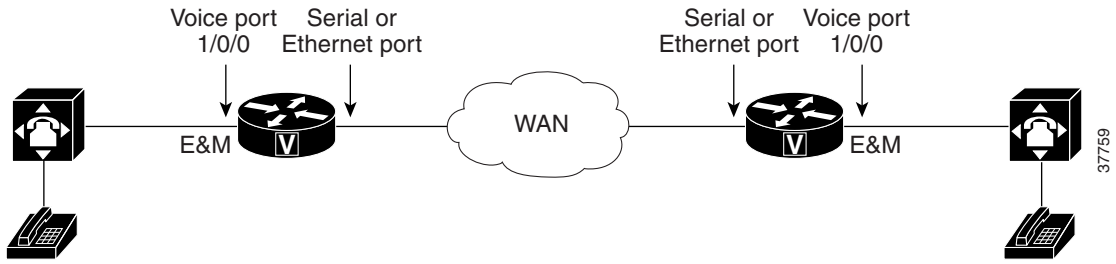


Figure 32 E&M Signaling Interfaces



FXS and FXO Interfaces

An FXS interface connects the router or access server to end-user equipment such as telephones, fax machines, or modems. The FXS interface supplies ring, voltage, and dial tone to the station and includes an RJ-11 connector for basic telephone equipment, keysets, and PBXs.

An FXO interface is used for trunk, or tie line, connections to a PSTN CO or to a PBX that does not support E&M signaling (when local telecommunications authority permits). This interface is of value for off-premise station applications. A standard RJ-11 modular telephone cable connects the FXO voice interface card to the PSTN or PBX through a telephone wall outlet.

FXO and FXS interfaces indicate on-hook or off-hook status and the seizure of telephone lines by one of two access signaling methods: loop start or ground start. The type of access signaling is determined by the type of service from the CO; standard home telephone lines use loop start, but business telephones can order ground start lines instead.

Loop-start is the more common of the access signaling techniques. When a handset is picked up (the telephone goes off-hook), this action closes the circuit that draws current from the telephone company CO and indicates a change in status, which signals the CO to provide dial tone. An incoming call is signaled from the CO to the handset by sending a signal in a standard on/off pattern, which causes the telephone to ring.

Loop-start has two disadvantages, however, that usually are not a problem on residential telephones but that become significant with the higher call volume experienced on business telephones. Loop-start signaling has no means of preventing two sides from seizing the same line simultaneously, a condition known as *glare*. Also, loop start signaling does not provide switch-side disconnect supervision for FXO calls. The telephony switch (the connection in the PSTN, another PBX, or key system) expects the router's FXO interface, which looks like a telephone to the switch, to hang up the calls it receives through its FXO port. However, this function is not built into the router for received calls; it only operates for calls originating from the FXO port.

Another access signaling method used by FXO and FXS interfaces to indicate on-hook or off-hook status to the CO is ground start signaling. It works by using ground and current detectors that allow the network to indicate off-hook or seizure of an incoming call independent of the ringing signal and allow for positive recognition of connects and disconnects. For this reason, ground start signaling is typically used on trunk lines between PBXs and in businesses where call volume on loop start lines can result in glare. See the "Disconnect Supervision Commands" section on page 118 and "FXO Supervisory Disconnect Tone Commands" section on page 121 for voice port commands that configure additional recognition of disconnect signaling.

In most cases, the default voice port command values are sufficient to configure FXO and FXS voice ports.

E&M Interfaces

Trunk circuits connect telephone switches to one another; they do not connect end-user equipment to the network. The most common form of analog trunk circuit is the E&M interface, which uses special signaling paths that are separate from the trunk's audio path to convey information about the calls. The signaling paths are known as the *E-lead* and the *M-lead*. The name *E&M* is thought to derive from the phrase *Ear* and *Mouth* or *rEceive* and *transMit* although it could also come from *Earth* and *Magnet*. The history of these names dates back to the days of telegraphy, when the CO side had a key that grounded the E circuit, and the other side had a sounder with an electromagnet attached to a battery. Descriptions such as *Ear* and *Mouth* were adopted to help field personnel determine the direction of a signal in a wire. E&M connections from routers to telephone switches or to PBXs are preferable to FXS/FXO connections because E&M provides better answer and disconnect supervision.

Like a serial port, an E&M interface has a data terminal equipment/data communications equipment (DTE/DCE) type of reference. In the telecommunications world, the *trunking* side is similar to the DCE, and is usually associated with CO functionality. The router acts as this side of the interface. The other side is referred to as the *signaling* side, like a DTE, and is usually a device such as a PBX. Five distinct physical configurations for the signaling part of the interface (Types I-V) use different methods to signal on-hook/off-hook status, as shown in Table 8. Cisco voice implementation supports E&M Types I, II, III, and V.

The physical E&M interface is an RJ-48 connector that connects to PBX trunk lines, which are classified as either two-wire or four-wire. This refers to whether the audio path is full duplex on one pair of wires (two-wire) or on two pair of wires (four-wire). A connection may be called a four-wire E&M circuit although it actually has six to eight physical wires. It is an analog connection although an analog E&M circuit may be emulated on a digital line. For more information on digital voice port configuration of E&M signaling, see the "DS0 Groups on Digital T1/E1 Voice Ports" section on page 106.

PBXs built by different manufacturers can indicate on-hook/off-hook status and telephone line seizure on the E&M interface by using any of three types of access signaling that are as follows:

- Immediate-start is the simplest method of E&M access signaling. The calling side seizes the line by going off-hook on its E-lead and sends address information as dual-tone multifrequency (DTMF) digits (or as dialed pulses on Cisco 2600 series routers and Cisco 3600 series routers) following a short, fixed-length pause.
- Wink-start is the most commonly used method for E&M access signaling, and is the default for E&M voice ports. Wink-start was developed to minimize glare, a condition found in immediate-start E&M, in which both ends attempt to seize a trunk at the same time. In wink-start, the calling side seizes the line by going off-hook on its E-lead, then waits for a short temporary off-hook pulse, or "wink," from the other end on its M-lead before sending address information. The switch interprets the pulse as an indication to proceed and then sends the dialed digits as DTMF or dialed pulses.
- In delay-dial signaling, the calling station seizes the line by going off-hook on its E-lead. After a timed interval, the calling side looks at the status of the called side. If the called side is on-hook, the calling side starts sending information as DTMF digits; otherwise, the calling side waits until the called side goes on-hook and then starts sending address information.

Table 8 E&M Wiring and Signaling Methods

E&M Type	E-Lead Configuration	M-Lead Configuration	Signal Battery Lead Configuration	Signal Ground Lead Configuration
I	Output, relay to ground	Input, referenced to ground	—	—
II	Output, relay to SG	Input, referenced to ground	Feed for M, connected to –48V	Return for E, galvanically isolated from ground
III	Output, relay to ground	Input, referenced to ground	Connected to –48V	Connected to ground
V	Output, relay to ground	Input, referenced to –48V	—	—

Analog Voice Ports Configuration Task List

Analog voice port interfaces connect routers in packet-based networks to analog two-wire or four-wire analog circuits in telephony networks. Two-wire circuits connect to analog telephone or fax devices, and four-wire circuits connect to PBXs. Typically, connections to the PSTN CO are made with digital interfaces.

This section describes how to configure analog voice ports and covers the following topics:

- Configuring Codec Complexity for Analog Voice Ports on the Cisco MC3810 with High-Performance Compression Modules, page 81
- Configuring Basic Parameters on Analog FXO, FXS, or E&M Voice Ports, page 82
- Configuring Analog Telephone Connections on Cisco 803 and 804 Routers, page 86

Three other sections later in the chapter provide help with fine-tuning and troubleshooting:

- Fine-Tuning Analog and Digital Voice Ports, page 114
- Verifying Analog and Digital Voice-Port Configurations, page 133
- Troubleshooting Analog and Digital Voice Port Configurations, page 144

Prerequisites for Configuring Analog Voice Ports

- Obtain two- or four-wire line service from your service provider or from a PBX.
- Complete your company's dial plan.
- Establish a working telephony network based on your company's dial plan.
- Install at least one other network module or WAN interface card to provide the connection to the network LAN or WAN.
- Establish a working IP and Frame Relay or ATM network. For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*, Release 12.2.
- Install appropriate voice processing and voice interface hardware on the router. See the "Configuring Platform-Specific Analog Voice Hardware" section on page 79.

Preparing to Configure Analog Voice Ports

Before configuring an analog voice port, assemble the following information about the telephony connection that the voice port will be making. If connecting to a PBX, it is important to understand the PBX's wiring scheme and timing parameters. This information should be available from your PBX vendor or the reference manuals that accompany your PBX.

- Telephony signaling interface: FXO, FXS, or E&M
- Locale code (usually the country) for call progress tones
- If FXO, type of dialing: DTMF (touch-tone) or pulse
- If FXO, type of start signal: loop-start or ground-start
- If E&M, type: I, II, III, or V
- If E&M, type of line: two-wire or four-wire
- If E&M, type of start signal: wink, immediate, delay-dial

Table 9 should help you determine which hardware and configuration instructions are appropriate for your situation. Table 10 on page 78 shows slot and port numbering, which differs for each of the voice-enabled routers. More current information may be available in the release notes that accompany the Cisco IOS software you are using.

Table 9 Analog Voice Port Configurations

Telephony Signaling Interface	Router Platform	Voice Hardware Required	Section Containing Voice Port Configuration Instructions
End user: telephone or fax	Cisco 803 Cisco 804	—	"Configuring Analog Telephone Connections on Cisco 803 and 804 Routers"
FXO	Cisco 1750 Cisco 2600 series Cisco 3600 series	VIC-2FXO, VIC-2FXO-EU	"Configuring Basic Parameters on Analog FXO, FXS, or E&M Voice Ports"
	Cisco MC3810	MC3810-AVM6 MC3810-APM-FXO	
FXS	Cisco 1750 Cisco 2600 series Cisco 3600 series	VIC-2FXS	
	Cisco MC3810	MC3810-AVM6 MC3810-APM-FXS	
E&M	Cisco 1750 Cisco 2600 series Cisco 3600 series	VIC-2E/M	
	Cisco MC3810	MC3810-AVM6 MC3810-APM-EM	

Table 10 Analog Voice Slot/Port Designations

Router Platform	Voice Hardware	Chassis Slot Numbers	Voice NM Slot Numbers	Voice Port Numbers
Cisco 803, 804	Analog POTS	—	—	—
Cisco 1750	Analog VIC	0 to 1	—	0 to 1
Cisco 2600 series	Voice/fax network module with two-port VIC	Varies, based on router	1	0 to 1
Cisco 3600 series	Voice/fax network module with two-port voice over interface cards (VICs)	1	3620: 0 to 1 3640: 0 to 3 3660: 1 to 6	0 to 1
Cisco MC3810	Analog voice module (AVM)	1	—	1 to 6

Configuring Platform-Specific Analog Voice Hardware

This section describes the general types of analog voice port hardware available for the router platforms included in this chapter:

Note For current information about supported hardware, see the release notes for the platform and Cisco IOS release being used.

Cisco 800 Series Routers

Cisco 803 and Cisco 804 routers support data and voice applications. The data applications on these routers are implemented through the ISDN port, and the voice applications are implemented with ISDN Basic Rate Interface (BRI) through the telephone ports. If a Cisco 803 or 804 router is being used, connect two devices, such as an analog touch-tone telephone, fax machine, or modem through two fixed telephone ports, the gray PHONE 1 and PHONE 2 ports that have RJ-11 connectors. Each device is connected to basic telephone services through the ISDN line.

For more information, refer to the *Cisco 800 Series Routers Hardware Installation Guide*.

Cisco 1750 Modular Router

The Cisco 1750 modular router provides Voice over IP (VoIP) functionality and can carry voice traffic (for example, telephone calls and faxes) over an IP network. To make a voice connection, the router must have a supported VIC installed. The Cisco 1750 router supports two slots for either WAN interface cards (WICs) or VICs and supports one VIC-only slot. For analog connections, two-port VICs are available to support FXO, FXS, and E&M signaling. VICs provide direct connections to telephone equipment (analog phones, analog fax machines, key systems, or PBXs) or to a PSTN.

For more information, refer to the *Cisco 1750 Voice-over-IP Quick Start Guide*.

Cisco 2600 Series and Cisco 3600 Series Routers

The Cisco 2600 and 3600 series routers are modular, multifunction platforms that combine dial access, routing, local area network-to-local area network (LAN) services, and multiservice integration of voice, video, and data in the same device.

Voice network modules installed in Cisco 2600 series or Cisco 3600 series routers convert telephone voice signals into data packets that can be transmitted over an IP network. The voice network modules have no connectors; VICs installed in the network modules provide connections to the telephone equipment or network. VICs work with existing telephone and fax equipment and are compatible with H.323 standards for audio and video conferencing.

The Cisco 2600 series router can house one network module. In the Cisco 3600 series, the Cisco 3620 router has slots for up to two network modules; the Cisco 3640 router has slots for up to four network modules; and the Cisco 3660 router has slots for up to six network modules. (Typically, one of the slots is used for LAN connectivity.)

For analog telephone connections, low-density voice/fax network modules that contain either one or two VIC slots are installed in the network module slots. Each VIC is specific to a particular telephone signaling interface (FXS, FXO, or E&M); therefore, the VIC determines the type of signaling on that module.

For more information, refer to the following:

- *Cisco 2600 Series Hardware Installation Guide*
- *Cisco 3600 Series Hardware Installation Guide*
- *Cisco Network Module Hardware Installation Guide*

Cisco MC3810 Multiservice Concentrator

To support analog voice circuits, a Cisco MC3810 multiservice concentrator must be equipped with an AVM, which supports six analog voice ports. By installing specific signaling modules known as analog personality modules (APMs), the analog voice ports may be equipped for the following signaling types in various combinations: FXS, FXO, and E&M. For FXS, the analog voice ports use an RJ-11 connector interface to connect to analog telephones or fax machines (two-wire) or to a key system (four-wire). For FXO, the analog voice ports use an RJ-11 physical interface to connect to a CO trunk. For E&M connections, the analog voice ports use an RJ-1CX physical interface to connect to an analog PBX (two-wire or four-wire).

Optional high-performance voice compression modules (HCMs) can replace standard voice compression modules (VCMs) to operate according to the voice compression coding algorithm (codec) specified when the Cisco MC3810 concentrator is configured. The HCM2 provides four voice channels at high codec complexity and eight channels at medium complexity. The HCM6 provides 12 voice channels at high complexity and 24 channels at medium complexity. One or two HCMs can be installed in a Cisco MC3810 multiservice concentrator, but an HCM may not be combined with a VCM in one chassis.

For more information, refer to the *Cisco MC3810 Multiservice Concentrator Hardware Installation Guide.*

Note For current information about supported hardware, see the release notes for the platform and Cisco IOS release being used.

Configuring Codec Complexity for Analog Voice Ports on the Cisco MC3810 with High-Performance Compression Modules

The term *codec* stands for *coder-decoder*. A codec is a particular method of transforming analog voice into a digital bit stream (and vice versa) and also refers to the type of compression used. Several different codecs have been developed to perform these functions, and each one is known by the number of the International Telecommunication Union-Telecommunication Standardization Sector (ITU-T) standard in which it is defined. For example, two common codecs are the G.711 and the G.729 codecs. The various codecs use different algorithms to encode analog voice into digital bit-streams and have different bit rates, frame sizes, and coding delays associated with them. The codecs also differ in the amount of perceived voice quality they achieve. Specialized hardware and software in the digital signal processors (DSPs) perform codec transformation and compression functions, and different DSPs may offer different selections of codecs.

Select the same type of codec as the one that is used at the other end of the call. For instance, if a call was coded with a G.729 codec, it must be decoded with a G.729 codec. Codec choice is configured on dial peers. For more information, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide.

Codec complexity refers to the amount of processing power that a codec compression technique requires: some require more processing power than others. Codec complexity affects call density, which is the number of calls that can take place on the DSP interfaces, which can be HCMs, port adapter DSP farms, or voice cards, depending on the type of router (in this case, the Cisco MC3810 multiservice concentrator). The greater the codec complexity, the fewer the calls that can be handled.

Codec complexity is either medium or high. The difference between medium- and high-complexity codecs is the amount of CPU power necessary to process the algorithm and, therefore, the number of voice channels that can be supported by a single DSP. All medium-complexity codecs can also be run in high-complexity mode, but fewer (usually half as many) channels will be available per DSP.

For details on the number of calls that can be handled simultaneously using each of the codec standards, refer to the entries for the **codec** and **codec complexity** commands in the *Cisco IOS Voice, Video, and Fax Command Reference*.

On a Cisco MC3810 concentrator, only a single codec complexity setting is used, even when two HCMs are installed. The value that is specified in this task affects the choice of codecs available when the **codec** dial-peer configuration command is configured. See the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide.

Note On the Cisco MC3810 with high-performance compression modules, check the DSP voice channel activity with the **show voice dsp** command. If any DSP voice channels are in the busy state, the codec complexity cannot be changed. When all the DSP channels are in the idle state, changes can be made to the codec complexity selection.

To configure codec complexity on the Cisco MC3810 multiservice concentrator using HCMs, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show voice dsp`	Checks the DSP voice channel activity. If any DSP voice channels are in the busy state, the codec complexity cannot be changed. When all the DSP channels are in the idle state, continue to Step 2.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# voice-card 0`	Enters voice-card configuration mode and specifies voice card 0.
Step 4	`Router(config-voicecard)# codec complexity {high | medium}`	(For analog voice ports) Specifies codec complexity based on the codec standard being used. This setting restricts the codecs available in dial peer configuration. All voice cards in a router must use the same codec complexity setting. The keywords are as follows: • **high**—Specifies two voice channels encoded in any of the following formats: G.711ulaw, G.711alaw, G.723.1(r5.3), G.723.1 Annex A(r5.3), G.723.1(r6.3), G.723.1 Annex A(r6.3), G.726(r16), G.726(r24), G.726(r32), G.728, G.729, G.729 Annex B, and fax relay. • **medium**—(default) Specifies four voice channels encoded in any of the following formats: G.711ulaw, G.711alaw, G.726(r16), G.726(r24), G.726(r32), G.729 Annex A, G.729 Annex B with Annex A, and fax relay. **Note** If two HCMs are installed, this command configures both HCMs at once.

Configuring Basic Parameters on Analog FXO, FXS, or E&M Voice Ports

This section describes commands for basic analog voice port configuration. All the data recommended in the "Preparing to Configure Analog Voice Ports" section on page 77 should be gathered before starting this procedure.

If configuring a Cisco MC3810 multiservice concentrator that has HCMs, codec complexity should also be configured, following the steps in the "Configuring Codec Complexity for Analog Voice Ports on the Cisco MC3810 with High-Performance Compression Modules" section on page 81.

Note If you have a Cisco MC3810 multiservice concentrator or Cisco 3660 router, the **compand-type a-law** command must be configured on the analog ports only. The Cisco 2660, 3620, and 3640 routers do not require the configuration of th **compand-type a-law** command, however, if you request a list of commands, the **compand-type a-law** command will display.

In addition to the basic voice port parameters described in this section, there are commands that allow voice port configurations to be fine tuned. In most cases, the default values for fine-tuning commands are sufficient for establishing FXO and FXS voice port configurations. E&M voice ports are more likely to require some configuration. If it is necessary to change some of the voice port values to improve voice quality or to match parameters on proprietary PBXs to which you are connecting, use the commands in the current section and also in the "Fine-Tuning Analog and Digital Voice Ports" section on page 114.

After the voice-port has been configured, make sure that the ports are operational by following the steps described in the following sections:

- Verifying Analog and Digital Voice-Port Configurations, page 133
- Troubleshooting Analog and Digital Voice Port Configurations, page 144

For more information on these and other voice port commands, see the *Cisco IOS Voice, Video, and Fax Command Reference*.

Note The commands, keywords, and arguments that you are able to use may differ slightly from those presented here, based on your platform, Cisco IOS release, and configuration. When in doubt, use Cisco IOS command help (**command ?**) to determine the syntax choices that are available.

To configure basic analog voice port parameters on Cisco 1750, Cisco 2600 series, Cisco 3600 series, and Cisco MC3810 routers, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco 1750 and MC3810**
	`Router(config)# voice-port slot/port`
	
	Cisco 2600 and 3600 series
	`Router(config)# voice-port slot/subunit/port`
		Enters voice-port configuration mode.
		The arguments are as follows:
		• *slot*—Specifies the number of the router slot where the voice network module is installed (Cisco 2600 and Cisco 3600 series routers) or the router slot number where the analog voice module is installed (Cisco MC3810 multiservice concentrator).
		• *port*—Indicates the voice port. Valid entries are 0 or 1.
		• *subunit*—Specifies the location of the VIC.
		Note The slash must be entered between *slot* and *port*.
		Valid entries vary by router platform; see Table 10 on page 78 or enter the **show voice port summary** command for available values.

Step 2	**FXO or FXS** `Router(config-voiceport)# signal {loop-start | ground-start}`	Selects the access signaling type to match that of the telephony connection you are making. The keywords are as follows: • **loop-start**—(default) Uses a closed circuit to indicate off-hook status; used for residential loops. • **ground-start**—Uses ground and current detectors; preferred for PBXs and trunks.

	Command	Purpose
	E&M `Router(config-voiceport)# signal {wink-start | immediate-start | delay-dial}`	The keywords are as follows: • **wink-start**—(default) Indicates that the calling side seizes the line, then waits for a short off-hook *wink* from the called side before proceeding. • **immediate-start**—Indicates that the calling side seizes the line and immediately proceeds; used for E&M tie trunk interfaces. • **delay-dial**—Indicates that the calling side seizes the line and waits, then checks to determine whether the called side is on-hook before proceeding; if not, it waits until the called side is on-hook before sending digits. Used for E&M tie trunk interfaces. **Note** Configuring the **signal** keyword for one voice port on a Cisco 2600 or 3600 series router VIC changes the signal value for both ports on the VIC.
Step 3	`Router(config-voiceport)# cptone locale`	Selects the two-letter locale for the voice call progress tones and other locale-specific parameters to be used on this voice port. Cisco routers comply with the ISO 3166 locale name standards. To see valid choices, enter a question mark (?) following the **cptone** command. The default is **us**.
Step 4	`Router(config-voiceport)# dial-type {dtmf | pulse}`	(FXO only) Specifies the dialing method for outgoing calls.
Step 5	`Router(config-voiceport)# operation {2-wire | 4-wire}`	(E&M only) Specifies the number of wires used for voice transmission at this interface (the audio path only, not the signaling path). The default is 2-wire.
Step 6	`Router(config-voiceport)# type {1 | 2 | 3 | 5}`	(E&M only) Specifies the type of E&M interface to which this voice port is connecting. See Table 8 on page 76 for an explanation of E&M types. The default is 1.

Step 7	**Cisco 1750 Router and 2600 and 3600 Series Routers**
	`Router(config-voiceport)# ring frequency {25 | 50}`
	Cisco MC3810 Multiservice Concentrator
	`Router(config-voiceport)# ring frequency {20 | 30}`
		(FXS only) Selects the ring frequency, in hertz, used on the FXS interface. This number must match the connected telephony equipment and may be country-dependent. If not set properly, the attached telephony device may not ring or it may buzz.
		The keyword default is 25 on the Cisco 1750 router, 2600 and 3600 series routers; and 20 on the Cisco MC3810 multiservice concentrator.

	Command	Purpose
Step 8	`Router(config-voiceport)# ring number number`	(FXO only) Specifies the maximum number of rings to be detected before an incoming call is answered by the router. The default is 1.
Step 9	`Router(config-voiceport)# ring cadence {[pattern01 | pattern02 | pattern03 | pattern04 | pattern05 | pattern06 | pattern07 | pattern08 | pattern09 | pattern10 | pattern11 | pattern12] | [define pulse interval]}`	(FXS only) Specifies an existing pattern for ring, or it defines a new one. Each pattern specifies a ring-pulse time and a ring-interval time. The keywords and arguments are as follows: • **pattern01** through **pattern12** name pre-set ring cadence patterns. Enter **ring cadence ?** to see ring pattern explanations. • **define** *pulse interval* specifies a user-defined pattern: *pulse* is a number (one or two digits, from 1 to 50) specifying ring pulse (on) time in hundreds of milliseconds, and *interval* is a number (one or two digits from 1 to 50) specifying ring interval (off) time in hundreds of milliseconds. The default is the pattern specified by the cptone locale that has been configured.
Step 10	`Router(config-voiceport)# description string`	Attaches a text string to the configuration that describes the connection for this voice port. This description appears in various displays and is useful for tracking the purpose or use of the voice port. The *string* argument is a character string from 1 to 255 characters in length. The default is that there is no text string (describing the voice port) attached to the configuration.
Step 11	`Router(config-voiceport)# no shutdown`	Activates the voice port. If a voice port is not being used, shut the voice port down with the **shutdown** command.

Configuring Analog Telephone Connections on Cisco 803 and 804 Routers

Multiple devices (analog telephone, fax machine, or modem) can be connected to a Cisco 803 or 804 telephone port. The number of devices that can be connected depends on the ringer equivalent number (REN) of each device that is to be connected. (The REN can usually be found on the bottom of a device.) The REN of the router telephone port is 5, so if the REN of each device to be connected is 1, a maximum of five devices can be connected to that particular telephone port.

These routers support touch-tone analog telephones only; they do not support rotary telephones.

To configure standard features for analog telephone connections on Cisco 803 and 804 routers, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# pots country` *country*	Specifies the country to use for country-specific default settings for physical characteristics. Enter **pots country ?** for a list of supported countries and the codes to enter. A default country is not defined.
Step 2	`Router(config)# pots line-type {type1 | type2 | type3}`	(Optional) Specifies the impedance of telephones, fax machines, or modems connected to a Cisco 800 series router. The keywords are as follows: • **type1**—Specifies the resistance used for the POTS connection, typically 600 ohms. • **type2**—Specifies the resistance used for the POTS connection, typically 900 ohms. • **type3**—Specifies the resistance used for the POTS connection, typically 300/400 ohms. The default depends on the country chosen in the **pots country** command.
Step 3	`Router(config)# pots dialing-method {overlap | enblock}`	(Optional) Specifies how the router collects and sends digits dialed on connected telephones, fax machines, or modems. The keywords are as follows: • **overlap**—Tells the router to send each digit dialed in a separate message. • **enblock**—Tells the router to collect all digits dialed and to send the digits in one message. The default depends on the country chosen in the **pots country** command.

	Command	Purpose
Step 4	`Router(config)# pots disconnect-supervision {osi | reversal}`	(Optional) Specifies how the router notifies the connected telephones, fax machines, or modems when the calling party has disconnect. The keywords are as follows: • **osi**—(open switching interval) Specifies the duration for which DC voltage applied between tip and ring conductors of a telephone port is removed. • **reversal**—Specifies the polarity reversal of the tip and ring conductors of a telephone port. The default depends on the country chosen in the **pots country** command.
Step 5	`Router(config)# pots encoding {alaw | ulaw}`	(Optional) Specifies the pulse code modulation (PCM) encoding scheme for telephones, fax machines, or modems connected to a Cisco 800 series router. The keywords are as follows: • **alaw**—Specifies the ITU-T PCM encoding scheme used to represent analog voice samples as digital values. • **ulaw**—Specifies the North American PCM encoding scheme used to represent analog voice samples as digital values. The default depends on the country chosen in the **pots country** command.
Step 6	`Router(config)# pots tone-source {local | remote}`	(Optional) Specifies the source of dial, ringback, and busy tones for telephones, fax machines, or modems connected to a Cisco 800 series router. The keywords are as follows: • **local**—(default) Specifies that the router supplies the tones. • **remote**—Specifies that the telephone switch supplies the tones.
Step 7	`Router(config)# pots ringing-freq {20Hz | 25Hz | 50Hz}`	(Optional) Specifies the frequency at which telephones, fax machines, or modems connected to a Cisco 800 series router ring. The keywords are as follows: • **20Hz**—Indicates that connected devices ring at 20 Hz. • **25Hz**—Indicates that connected devices ring at 25 Hz. • **50Hz**—Indicates that connected devices ring at 50 Hz. The default depends on the country chosen in the **pots country** command.

	Command	Purpose
Step 8	`Router(config)# pots disconnect-time` *interval*	(Optional) Specifies the interval at which the disconnect method is applied if connected telephones, fax machines, or modems fail to detect that a calling party has disconnected. The *interval* argument is the number of milliseconds of the interval and ranges from 50 to 2000. The default depends on the country chosen in the **pots country** command.
Step 9	`Router(config)# pots silence-time` *seconds*	(Optional) Specifies the interval of silence after a calling party disconnects. The *seconds* argument is the number of seconds of the interval and ranges from 0 to 10. The default depends on the country chosen in the **pots country** command.
Step 10	`Router(config)# pots distinctive-ring-guard-time` *milliseconds*	(Optional) Specifies the delay after which a telephone port can be rung after a previous call is disconnected. The *milliseconds* argument is the number of milliseconds of the delay and ranges from 0 to 1000. The default depends on the country chosen in the **pots country** command.

Verifying Analog Telephone Connections on Cisco 803 and 804 Routers

After configuring analog telephone connections, perform the following steps to verify proper operation:

Step 1 Pick up the handset of an attached telephony device and check for a dial tone.

Step 2 Review the configuration using the **show pots status** command, which displays settings of physical characteristics and other information on telephone interfaces.

```
Router# show pots status

POTS Global Configuration:
 Country: United States
  Dialing Method: Overlap, Tone Source: Remote, CallerId Support: YES
  Line Type: 600 ohm, PCM Encoding: u-law, Disc Type: OSI,
  Ringing Frequency: 20Hz, Distinctive Ring Guard timer: 0 msec
  Disconnect timer: 1000 msec, Disconnect Silence timer: 5 sec
  TX Gain: 6dB, RX Loss: -6dB,
  Filter Mask: 6F
  Adaptive Cntrl Mask: 0
POTS PORT: 1
  Hook Switch Finite State Machine:
   State: On Hook, Event: 0
   Hook Switch Register: 10, Suspend Poll: 0
  CODEC Finite State Machine
   State: Idle, Event: 0
  Connection: None, Call Type: Two Party, Direction: Rx only
  Line Type: 600 ohm, PCM Encoding: u-law, Disc Type: OSI,
  Ringing Frequency: 20Hz, Distinctive Ring Guard timer: 0 msec
  Disconnect timer: 1000 msec, Disconnect Silence timer: 5 sec
  TX Gain: 6dB, RX Loss: -6dB,
```

```
  Filter Mask: 6F
  Adaptive Cntrl Mask: 0
  CODEC Registers:
   SPI Addr: 2, DSLAC Revision: 4
   SLIC Cmd: 0D, TX TS: 00, RX TS: 00
   Op Fn: 6F, Op Fn2: 00, Op Cond: 00
   AISN: 6D, ELT: B5, EPG: 32 52 00 00
   SLIC Pin Direction: 1F
  CODEC Coefficients:
   GX: A0 00
   GR: 3A A1
   Z: EA 23 2A 35 A5 9F C2 AD 3A AE 22 46 C2 F0
   B: 29 FA 8F 2A CB A9 23 92 2B 49 F5 37 1D 01
   X: AB 40 3B 9F A8 7E 22 97 36 A6 2A AE
   R: 01 11 01 90 01 90 01 90 01 90 01 90
   GZ: 60
  ADAPT B: 91 B2 8F 62 31
 CSM Finite State Machine:
   Call 0 - State: idle, Call Id: 0x0
    Active: no
   Call 1 - State: idle, Call Id: 0x0
    Active: no
   Call 2 - State: idle, Call Id: 0x0
    Active: no
 POTS PORT: 2
   Hook Switch Finite State Machine:
    State: On Hook, Event: 0
    Hook Switch Register: 20, Suspend Poll: 0
   CODEC Finite State Machine:
    State: Idle, Event: 0
    Connection: None, Call Type: Two Party, Direction: Rx only
    Line Type: 600 ohm, PCM Encoding: u-law, Disc Type: OSI,
    Ringing Frequency: 20Hz, Distinctive Ring Guard timer: 0 mse
  Disconnect timer: 1000msec,Disconnect Silence timer: 5 sec
   TX Gain: 6dB, RX Loss: -6dB,
   Filter Mask: 6F
   Adaptive Cntrl Mask: 0
 CODEC Registers:
   SPI Addr: 3, DSLAC Revision: 4
   SLIC Cmd: 0D, TX TS: 00, RX TS: 00
   Op Fn: 6F, Op Fn2: 00, Op Cond: 00
   AISN: 6D, ELT: B5, EPG: 32 52 00 00
   SLIC Pin Direction: 1F
 CODEC Coefficients:
   GX: A0 00
   GR: 3A A1
   Z: EA 23 2A 35 A5 9F C2 AD 3A AE 22 46 C2 F0
   B: 29 FA 8F 2A CB A9 23 92 2B 49 F5 37 1D 01
   X: AB 40 3B 9F A8 7E 22 97 36 A6 2A AE
   R: 01 11 01 90 01 90 01 90 01 90 01 90
   GZ: 60
  ADAPT B: 91 B2 8F 62 31
 CSM Finite State Machine:
   Call 0 - State: idle, Call Id: 0x0
    Active: no
   Call 1 - State: idle, Call Id: 0x0
    Active: no
   Call 2 - State: idle, Call Id: 0x0
    Active: no
Time Slot Control: 0
```

Troubleshooting Tip for Cisco 803 and 804 Routers

Check to ensure that all cables are securely connected.

Configuring Digital Voice Ports

The digital voice port commands discussed in this section configure channelized T1 or E1 connections; for information on ISDN connections, see "Configuring ISDN Interfaces for Voice" in this configuration guide.

The T1 or E1 lines that connect a telephony network to the digital voice ports on a router or access server contain channels for voice calls; a T1 line contains 24 full-duplex channels or *timeslots*, and an E1 line contains 30. The signal on each channel is transmitted at 64 kbps, a standard known as digital signal 0 (DS0); the channels are known as DS0 channels. The **ds0-group** command creates a logical voice port (a DS0 group) from some or all of the DS0 channels, which allows you to address those channels easily, as a group, in voice-port configuration commands.

Digital voice ports are found at the intersection of a packet voice network and a digital, circuit-switched telephone network. The digital voice port interfaces that connect the router or access server to T1 or E1 lines pass voice data and signaling between the packet network and the circuit-switched network.

Signaling is the exchange of information about calls and connections between two ends of a communication path. For instance, signaling communicates to the call's end points whether a line is idle or busy, whether a device is on-hook or off-hook, and whether a connection is being attempted. An end point can be a CO switch, a PBX, a telephony device such as a telephone or fax machine, or a voice-equipped router acting as a gateway. There are two aspects to consider about signaling on digital lines: one aspect is the actual information about line and device states that is transmitted, and the second aspect is the method used to transmit the information on the digital lines.

The actual information about line and device states is communicated over digital lines using signaling methods that emulate the methods used in analog circuit-switched networks: FXS, FXO, and E&M.

The method used to transmit the information describes the way that the emulated analog signaling is transmitted over digital lines, which may be *common-channel signaling* (CCS) or *channel-associated signaling* (CAS). CCS sends signaling information down a dedicated channel and CAS takes place within the voice channel itself. This chapter describes CAS signaling, which is sometimes called *robbed-bit signaling* because user bandwidth is *robbed* by the network for signaling. A bit is taken from every sixth frame of voice data to communicate on- or off-hook status, wink, ground start, dialed digits, and other information about the call.

In addition to setting up and tearing down calls, CAS provides the receipt and capture of dialed number identification (DNIS) and automatic number identification (ANI) information, which are used to support authentication and other functions. The main disadvantage of CAS signaling is its use of user bandwidth to perform these signaling functions.

For signaling to pass between the packet network and the circuit-switched network, both networks must use the same type of signaling. The voice ports on Cisco routers and access servers can be configured to match the signaling of most COs and PBXs, as explained in this chapter.

This section discusses the following topics:

Prerequisites for Configuring Digital Voice Ports

Digital T1 or E1 packet voice capability requires specific service, software, and hardware:

- Obtain T1 or E1 service from the service provider or from your PBX.
- Create your company's dial plan.
- Establish a working telephony network based on your company's dial plan.
- Establish a connection to the network LAN or WAN.
- Set up a working IP and Frame Relay or ATM network. For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*, Release 12.2.
- Install appropriate voice processing and voice interface hardware on the router. See the "Platform-Specific Digital Voice Hardware" section on page 94.
- (Cisco 2600 and 3600 series routers) For digital T1 packet voice trunk network modules, install Cisco IOS Release 12.0(5)XK, 12.0(7)T, 12.2(1), or a later release. The minimum DRAM memory requirements are as follows:
  - 32 MB, with one or two T1 lines
  - 48 MB, with three or four T1 lines
  - 64 MB, with five to ten T1 lines
  - 128 MB, with more than ten T1 lines

  The memory required for high-volume applications may be greater than that listed. Support for digital T1 packet voice trunk network modules is included in Plus feature sets. The IP Plus feature set requires 8 MB of Flash memory; other Plus feature sets require 16 MB.
- (Cisco 2600 and 3600 series routers) For digital E1 packet voice trunk network modules, install Cisco IOS Release 12.1(2)T, 12.2(1), or a later release. The minimum DRAM memory requirements are:
  - 48 MB, with one or two E1s
  - 64 MB, with three to eight E1s
  - 128 MB, with 9 to 12 E1s

  For high-volume applications, the memory required may be greater than these minimum values. Support for digital E1 packet voice trunk network modules is included in Plus feature sets. The IP Plus feature set requires 16 MB of Flash memory.
- (Cisco MC3810 concentrators) HCMs require Cisco IOS Release 12.0(7)XK or 12.1(2)T, 12.2(1), or a later release.
- (Cisco 7200 and 7500 series routers) For digital T1/E1 voice port adapters, install Cisco IOS Release 12.0(5)XE, 12.0(7)T, 12.2(1), or a later release. The minimum DRAM memory requirement to support T1/E1 high-capacity digital voice port adapters is 64 MB.

The memory required for high-volume applications may be greater than that listed. Support for T1/E1 high-capacity digital voice port adapters is included in Plus feature sets. The IP Plus feature set requires 16 MB of Flash memory.

Preparing Information to Configure Digital Voice Ports

Gather the following information about the telephony network connection that the voice port will be making:

- Line interface: T1 or E1
- Signaling interface: FXO, FXS, or E&M. If the interfaces are Primary Rate Interface (PRI) or BRI, see the "Configuring ISDN Interfaces for Voice" chapter in this configuration guide and *Cisco IOS Terminal Services Configuration Guide*.
- Line coding: AMI or B8ZS for T1, and AMI or HDB3 for E1
- Framing format: SF (D4) or ESF for T1, and CRC4 or no-CRC4 for E1
- Number of channels

Table 11 describes voice-port hardware configurations for various platforms. After the controllers have been configured, the **show voice port summary** command can also be used to determine available voice port numbers. If the **show voice port** command and a specific port number is entered, the default voice-port configuration for that port displays.

Table 11 Digital Voice Slot/Port Designations

Router Platform	Voice Hardware	Slot Number	Port Number
Cisco 2600 series	Digital T1/E1 Packet Voice Trunk Network Module (NM-HDV with VWIC-1MFT or VWIC-2MFT) One network module can be installed in a Cisco 2600 series router.	*slot* is the router location of the voice module. 1	*port* is the VWIC location in the network module. 0 to 1
Cisco 3600 series	Digital T1/E1 Packet Voice Trunk Network Module (NM-HDV with VWIC-1MFT or VWIC-2MFT) One network module can be installed in a Cisco 3620 router. A Cisco 3640 router can support three modules, and as many as six can be installed in a Cisco 3660 router.	*slot* is the router location of the voice module. 3620: 0 to 1 3640: 0 to 3 3660: 0 to 5	*port* is the VWIC location in the network module. 0 to 1

Table 11 Digital Voice Slot/Port Designations (continued)

Router Platform	Voice Hardware	Slot Number	Port Number
Cisco MC3810	• Digital voice module (DVM) • Voice compression module (VCM3 or VCM6) or • High-compression module (HCM2 or HCM6) VCM3 and VCM6 do not support codec complexity options.	1	—
Cisco AS5300	One Octal T1/E1 feature card (eight ports) or one Quad T1/E1 feature card (four ports) and one or two VFCs for voice and fax features.	—	*controller* is : Octal: 0 to 7 Quad: 0 to 3
Cisco AS5800	Up to four 12-port T1/E1 trunk cards and up to eight VFCs	*shelf* is 1 *slot* is 0 to 5	0 to 11
Cisco 7200 series	• Two-port T1/E1 enhanced digital voice port adapters • PA-VXC (high-capacity) • PA-VXB (moderate capacity) Port adapter slot 0 is reserved for the Fast Ethernet port on the I/O controller (if present).	Port adapter slot: from 1 to 4, or from 1 to 6	Interface port: 0 to 1
Cisco 7500 series	PA-VXB and PA-VXC on a VIP2 or VIP4 in Cisco 7500 series routers If the VIP is inserted in interface processor slot 3 and port adapter slot 0, then the addresses of the PA-VXB or PA-VXC are 3/0/0 or 3/0/1 (interface processor slot 3, port adapter slot 0, and interfaces 0 and 1).	Interface processor slot: 0 to 12 (depends on the number of slots in the router)	Port adapter slot: always 0 or 1 Interface port: 0 or 1

The following is **show voice port summary** sample output for a Cisco MC3810 multiservice concentrator:

```
Router# show voice port summary

IN      OUT
PORT   CH SIG-TYPE   ADMIN OPER STATUS   STATUS   EC
====== == ========== ===== ==== ======== ======== ==
0:17   18 fxo-ls     down  down idle     on-hook  y
0:18   19 fxo-ls     up    dorm idle     on-hook  y
0:19   20 fxo-ls     up    dorm idle     on-hook  y
0:20   21 fxo-ls     up    dorm idle     on-hook  y
0:21   22 fxo-ls     up    dorm idle     on-hook  y
0:22   23 fxo-ls     up    dorm idle     on-hook  y
0:23   24 e&m-imd    up    dorm idle     idle     y
```

Platform-Specific Digital Voice Hardware

This section briefly describes digital voice hardware on the following platforms:

- Cisco 2600 series and Cisco 3600 series routers
- Cisco MC3810 multiservice concentrator
- Cisco AS5300 universal access server
- Cisco AS5800 universal access server
- Cisco 7200 series and Cisco 7500 series routers

Note For current information about supported hardware, see the release notes for the platform and Cisco IOS release you are using.

Cisco 2600 Series and Cisco 3600 Series Routers

Digital voice hardware on Cisco 2600 series and Cisco 3600 series modular access routers includes the high-density voice (HDV) network module and the multiflex trunk (MFT) voice/WAN interface card (VWIC). When an HDV is used in conjunction with an MFT and packet voice DSP modules (PVDMs), the HDV module is also called a *digital packet voice trunk network module*. The digital T1 or E1 packet voice trunk network module supports T1 or E1 applications, including fractional use. The T1 version integrates a fully managed data service unit/channel service unit (DSU/CSU), and the E1 version includes a fully managed DSU. The digital T1 or E1 packet voice trunk network module provides per-channel T1 or E1 data rates of 64 or 56 kbps for WAN services (Frame Relay or leased line).

Digital T1 or E1 packet voice trunk network modules for Cisco 2600 and 3600 series routers allow enterprises or service providers, using the voice-equipped routers as customer premise equipment (CPE), to deploy digital voice and fax relay. These network modules receive constant bit-rate telephony information over T1 or E1 interfaces and convert that information to a compressed format so that it can be sent over a packet network. The digital T1 or E1 packet voice trunk network modules can connect either to a PBX (or similar telephony device) or to a CO to provide PSTN connectivity. One digital T1 or E1 packet voice trunk network module can be installed in a Cisco 2600 series router or in a Cisco 3620 router. A Cisco 3640 router can support three network modules, and a Cisco 3660 router can support up to six network modules.

The MFT VWICs that are used in the packet voice trunk network modules are available in one- and two-port configurations for T1 and for E1, and in two-port configurations with drop-and-insert capability for T1 and E1. MFTs support the following kinds of traffic:

- Data. As WICs for T1 or E1 applications, including fractional data line use, the T1 version includes a fully managed DSU/CSU, and the E1 version includes a fully managed DSU.
- Packet voice. As VWICs included with the digital T1 or E1 packet voice trunk network module to provide connections to PBXs and COs, the MFTs enable packet voice applications.
- Multiplexed voice and data. Some two-port T1 or E1 VWICs can provide drop-and-insert multiplexing services with integrated DSU/CSUs. For example, when used with a digital T1 packet voice trunk network module, drop-and-insert allows 64-kbps DS0 channels to be taken from one T1 and digitally cross-connected to 64-kbps DS0 channels on another T1. Drop and insert, sometimes called TDM cross-connect, uses circuit switching rather than the DSPs that VoIP technology employs. (Drop-and-insert is described in the "Configuring Trunk Connections and Trunk Conditioning Features" chapter in this configuration guide.)

The digital T1 or E1 packet voice trunk network module contains five 72-pin Single In-line Memory Module (SIMM) sockets or banks, numbered 0 through 4, for PVDMs. Each socket can be filled with a single 72-pin PVDM, and there must be at least one packet voice data module (PVDM-12) in the network module to process voice calls. Each PVDM holds three digital signal processors (DSPs), so with five PVDM slots populated, a total of 15 DSPs are provided. High-complexity codecs support two simultaneous calls on each DSP, and medium-complexity codecs support four calls on each DSP. A digital T1 or E1 packet voice trunk network module can support the following numbers of channels:

- When the digital T1 or E1 packet voice trunk network module is configured for high-complexity codec mode, up to six voice or fax calls can be completed per PVDM-12, using the following codecs: G.711, G.726, G.729, G729 Annex A (E1), G.729 Annex B, G.723.1, G723.1 Annex A (T1), G.728, and fax relay.
- When the digital T1 or E1 packet voice trunk network module is configured for medium-complexity codec mode, up to 12 voice or fax calls can be completed per PVDM-12, using the following codecs: G.711, G.726, G.729 Annex A, G.729 Annex B with Annex A, and fax relay.

For more information, refer to the following publications:

- *Cisco 2600 Series Hardware Installation Guide*
- *Cisco 3600 Series Hardware Installation Guide*
- *Cisco Network Module Hardware Installation Guide*
- Cisco IOS Release 12.0(7)T online document *Configuring 1- and 2-Port T1/E1 Multiflex Voice/WAN Interface Cards on Cisco 2600 and 3600 Series Routers*

Cisco MC3810 Multiservice Concentrator

To support a T1 or E1 digital voice interface, the Cisco MC3810 multiservice concentrator must be equipped with a digital voice interface card (DVM). The DVM interfaces with a digital PBX, channel bank, or video codec. It supports up to 24 channels of compressed digital voice at 8 kbps, or it can cross-connect channelized data from user equipment directly onto the router's trunk port for connection to a carrier network.

The DVM is available with a balanced interface using an RJ-48 connector or with an unbalanced interface using Bayonet-Neill-Concelman (BNC) connectors.

Optional HCMs can replace standard VCMs to operate according to the voice compression coding algorithm (codec) specified when the Cisco MC3810 multiservice concentrator is configured. The HCM2 provides 4 voice channels at high codec complexity and 8 channels at medium complexity. The

HCM6 provides 12 voice channels at high complexity and 24 channels at medium complexity. You can install one or two HCMs in a Cisco MC3810, but an HCM can not be combined with a VCM in the same chassis.

For more information, refer to the following publications:

- *Cisco MC3810 Multiservice Concentrator Hardware Installation Guide*
- *Overview of the Cisco MC3810 Series*
- *Configuring Cisco MC3810 Series Concentrators to Use High-Performance Compression Modules*

Cisco AS5300 Universal Access Server

The Cisco AS5300 Universal Access Server includes three expansion slots. One slot is for either an Octal T1/E1/PRI feature card (eight ports) or a Quad T1/E1/PRI feature card (four ports), and the other two can be used for voice/fax or modem feature cards. Because a single voice/fax feature card (VFC) can support up to 48 (T1) or 60 (E1) voice calls, the Cisco AS5300/Voice Gateway system can support a total of 96 or 120 simultaneous voice calls. The use of VFCs requires Cisco IOS release 12.0.2XH or later.

Cisco AS5300 VFCs are coprocessor cards, each with a powerful reduced instruction set computing (RISC) engine and dedicated, high-performance DSPs to ensure predictable, real-time voice processing. The design couples this coprocessor with direct access to the Cisco AS5300 routing engine for streamlined packet forwarding.

For more information, refer to the following publications:

- *Cisco AS5300 Chassis Installation Guide*
- *Cisco AS5300 Module Installation Guide*

Cisco AS5800 Universal Access Server

The Cisco AS5800 Universal Access Server consists of two primary system components: the Cisco 5814 dial shelf (DS), which holds channelized trunk cards and connects to the PSTN, and the Cisco 7206 router shelf (RS), which holds port adapters and connects to the IP backbone.

The dial shelf acts as the access concentrator by accepting and consolidating all types of remote traffic, including voice, dial-in analog and digital ISDN data, and industry-standard WAN and remote connection types. The dial shelf also contains controller cards voice feature cards, modem feature cards, trunk cards, and dial shelf interconnect cards.

One or two dial shelf controllers (DSCs) provide clock and power control to the dial shelf cards. Each DSC contains a block of logic that is referred to as the common logic and system clocks. This block of logic can use a variety of sources to generate the system timing, including an E1 or T1/T3 input signal from the BNC connector on the DSC's front panel. The configuration commands for the master clock specify the various clock sources and a priority for each source (see the "Clock Sources on Digital T1/E1 Voice Ports" section on page 102).

The Cisco AS5800 voice feature card is a multi-DSP coprocessing board and software package that adds VoIP capabilities to the Cisco AS5800 platform. The Cisco AS5800 voice feature card, when used with other cards such as LAN/WAN and modem cards, provides a gateway for up to 192 packetized voice/fax calls and 360 data calls per card. A Cisco AS5800 can support up to 1,344 voice calls in split-dial-shelf configuration with two 7206VXR router shelves.

For more information, refer to the following publications:

- *Cisco AS5800 Universal Access Server Operation, Administration, Maintenance, and Provisioning Guide*
- *Cisco AS5800 Access Server Hardware Installation Guide*

Cisco 7200 and Cisco 7500 Series Routers

Cisco 7200 and Cisco 7500 series routers support multimedia routing and bridging with a wide variety of protocols and media types. The Cisco 7000 family versatile interface processor (VIP) is based on a RISC engine optimized for I/O functions. To this engine are attached one or two port adapters or daughter boards, which provide the media-specific interfaces to the network. The network interfaces provide connections between the routers' peripheral component interconnect (PCI) buses and external networks. Port adapters can be placed in any available port adapter slot, in any desired combination.

T1/E1 high-capacity digital voice port adapters for Cisco 7200 and Cisco 7500 series routers allow enterprises or service providers, using the equipped routers as customer premise equipment, to deploy digital voice and fax relay. These port adapters receive constant bit-rate telephony information over T1/E1 interfaces and can convert that information to a compressed format for transmission as voice over IP (VoIP). Two types of digital voice port adapters are supported on Cisco 7200 and Cisco 7500 series routers: two-port high-capacity (up to 48 or 120 channels of compressed voice, depending on codec choice), and two-port moderate capacity (up to 24 or 48 channels of compressed voice). These single-width port adapters incorporate two universal ports configurable for either T1 or E1 connection, for use with high-performance digital signal processors (DSPs). Integrated CSU/DSUs, echo cancellation, and DS0 drop-and-insert functionality eliminate the need for external line termination devices and multiplexers.

For more information, refer to the following publications:

- *Cisco 7200 VXR Installation and Configuration Guide*
- *Cisco 7500 Series Installation and Configuration Guide*
- *Two-Port T1/E1 Moderate-Capacity and High-Capacity Digital Voice Port Adapter Installation and Configuration*

Note For current information about supported hardware, see the release notes for the platform and Cisco IOS release being used.

Configuring Basic Parameters on Digital T1/E1 Voice Ports

This section describes commands for basic digital voice port configuration. Make sure you have all the data recommended in the "Preparing Information to Configure Digital Voice Ports" section on page 92 before starting this procedure.

The basic steps for configuring digital voice ports are described in the next three sections. They are grouped by the configuration mode from which they are executed, as follows:

- Configuring Codec Complexity for Digital T1/E1 Voice Ports, page 98

  Codec complexity refers to the amount of processing power assigned to codec processing on a voice port. On most router platforms that support codec complexity, codec complexity is selected in voice card configuration mode, although it is selected in DSP interface mode on the Cisco 7200 and 7500 series. The value configured for codec complexity establishes the choice of codecs that are available on the dial peers. See the *Configuring Dial Plans, Dial Peers, and Digit Manipulation* chapter in this configuration guide for more information about configuring dial peers.

- Configuring Controller Settings for Digital T1/E1 Voice Ports, page 101

  Specific line characteristics must be configured to match those of the PSTN line that is being connected to the voice port. These are typically configured in controller configuration mode.

- Configuring Basic Voice Port Parameters for Digital T1/E1 Voice Ports, page 112

  Voice port configuration mode allows many of the basic voice call attributes to be configured to match those of the PSTN or PBX connection being made on this voice port.

In addition to the basic voice port parameters, there are additional commands that allow for the fine-tuning of the voice port configurations or for configuration of optional features. In most cases, the default values for these commands are sufficient for establishing voice port configurations. If it is necessary to change some of these parameters to improve voice quality or to match parameters in proprietary PBXs to which you are connecting, use the commands in the "Fine-Tuning Analog and Digital Voice Ports" section on page 114.

After voice port configuration, make sure the ports are operational by following the steps described in these sections:

- Verifying Analog and Digital Voice-Port Configurations, page 133
- Troubleshooting Analog and Digital Voice Port Configurations, page 144

For more information on voice port commands, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

Configuring Codec Complexity for Digital T1/E1 Voice Ports

On the Cisco 2600, 3600, 7200, and 7500 routers, codec complexity can be configured separately for each T1/E1 digital packet voice trunk network module or port adapter. On a Cisco MC3810 multiservice concentrator, only a single codec complexity setting is used, even when two HCMs are installed. The value specified in this task affects the choice of codecs available when the **codec** dial-peer configuration command is configured.

For details on the number of calls that can be handled simultaneously using each of the codec standards, refer to the entries for **codec** and **codec complexity** in the *Cisco IOS Voice, Video, and Fax Command Reference* and to platform-specific product literature.

For more information on codec complexity, see the "Configuring Codec Complexity for Analog Voice Ports on the Cisco MC3810 with High-Performance Compression Modules" section on page 81.

Two configuration task tables are shown below: one for the Cisco 2600 and 3600 series routers and the Cisco MC3810 concentrator, which use voice card configuration mode, and the second for the Cisco 7200 and 7500 series routers, which use DSP interface configuration mode.

Cisco 2600 and 3600 Series and Cisco MC3810

This procedure applies to voice ports on digital packet voice trunk network modules on Cisco 2600 series and Cisco 3600 series routers, and to voice ports on HCMs on Cisco MC3810 multiservice concentrators.

Note On Cisco 2600 and 3600 series routers with digital T1/E1 packet voice trunk network modules, codec complexity cannot be configured if DS0 groups are configured. Use the **no ds0-group** command to remove DS0 groups before configuring codec complexity.

Note On the Cisco MC3810 multiservice concentrator with high compression modules, check the DSP voice channel activity with the **show voice dsp** command. If any DSP voice channels are in the busy state, you cannot change the codec complexity. When all of the DSP channels are in the idle state, you can make changes to the codec complexity selection.

To configure codec complexity, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show voice dsp`	Checks the DSP voice channel activity. If any DSP voice channels are in the busy state, codec complexity cannot be changed. When all of the DSP channels are in the idle state, continue to Step 2.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# voice-card` *slot*	Enters voice card configuration mode for the card or cards in the slot specified. For the Cisco 2600 and 3600 series routers, the *slot* argument ranges from 0 to 5. For the Cisco MC3810 multiservice concentrator, *slot* must be 0.
Step 4	`Router(config-voicecard)# codec complexity {high | med}`	Specifies codec complexity based on the codec standard being used. This setting restricts the codecs available in dial peer configuration. All voice cards in a router must use the same codec complexity setting. The keywords are as follows: • **high**—(Optional) Specifies up to six voice or fax calls completed per PVDM-12, using the following codecs: G.711, G.726, G.729, G.729 Annex B, G.723.1, G.723.1 Annex A, G.728, and fax relay. • **med**—(Optional) Supports up to 12 voice or fax calls completed per PVDM-12, using the following codecs: G.711, G.726, G.729 Annex A, G.729 Annex B with Annex A, and fax relay. **T**he default is **med**. **Note** On the Cisco MC3810 multiservice concentrator, this command is valid only with one or more HCMs installed, and voice card 0 must be specified. If two HCMs are installed, this command configures both HCMs at once.

Cisco AS5300 Universal Access Server

Codec support on the Cisco AS5300 universal access server is determined by the capability list on the voice feature card, which defines the set of codecs that can be negotiated for a voice call. The capability list is created and populated when VCWare is unbundled and DSPWare is added to VFC Flash memory. The capability list does not indicate codec preference; it simply reports the codecs that are available. The session application decides which codec to use. Codec support is configured on dial peers rather than on voice ports; see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide.

Cisco AS5800 Universal Access Server

Selection of codec support on Cisco AS5800 access servers is made during dial peer configuration. See the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide.

Cisco 7200 Series and Cisco 7500 Series Routers

On Cisco 7200 series and Cisco 7500 series routers, codec complexity is configured on the DSP interface.

Note Check the DSP voice channel activity using the **show interfaces dspfarm** command. If any DSP voice channels are in the busy state, codec complexity cannot be changed. When all of the DSP channels are in the idle state, changes can be made to the codec complexity selection.

To configure the DSP interface, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show interfaces dspfarm`	Displays the DSP voice channel activity. If any DSP voice channels are in the busy state, codec complexity cannot be changed. When all of the DSP channels are in the idle state, continue to Step 2.
Step 2	`Router# configure terminal`	Enters global configuration mode.
Step 3	**Cisco 7200 series** `Router(config)# dspint dspfarm` *slot/port* **Cisco 7500 series** `Router(config)# dspint dspfarm` *slot/port-adapter/port*	Enters DSP interface configuration mode. The arguments are as follows: • *slot/port*—Specifies the slot and port numbers of the interface. • *adapter/port*—Specifies the adapter and port numbers of the interface.

	Command	Purpose
Step 4	`Router(config-dspfarm)# codec {high | med}`	Specifies the codec complexity based on the codec standard being used. The keyword specified for **codec** affects the choice of codecs available when the **codec** dial-peer configuration command is used. The keywords are as follows: • **high**—Supports two voice channels encoded in any of the following formats: G.711, G.726, G.729, G.729 Annex B, G.723.1, G.723.1 Annex A, G.728, and fax relay. • **med**—(default) Supports up to four calls using the following codecs: G.711, G.726, G.729 Annex A, G.729 Annex B with Annex A, and fax relay.
Step 5	`Router(config-dspfarm)# description`	Enters a string to include descriptive text about this DSP interface connection. This information is displayed in the output for **show** commands and does not affect the operation of the interface in any way.

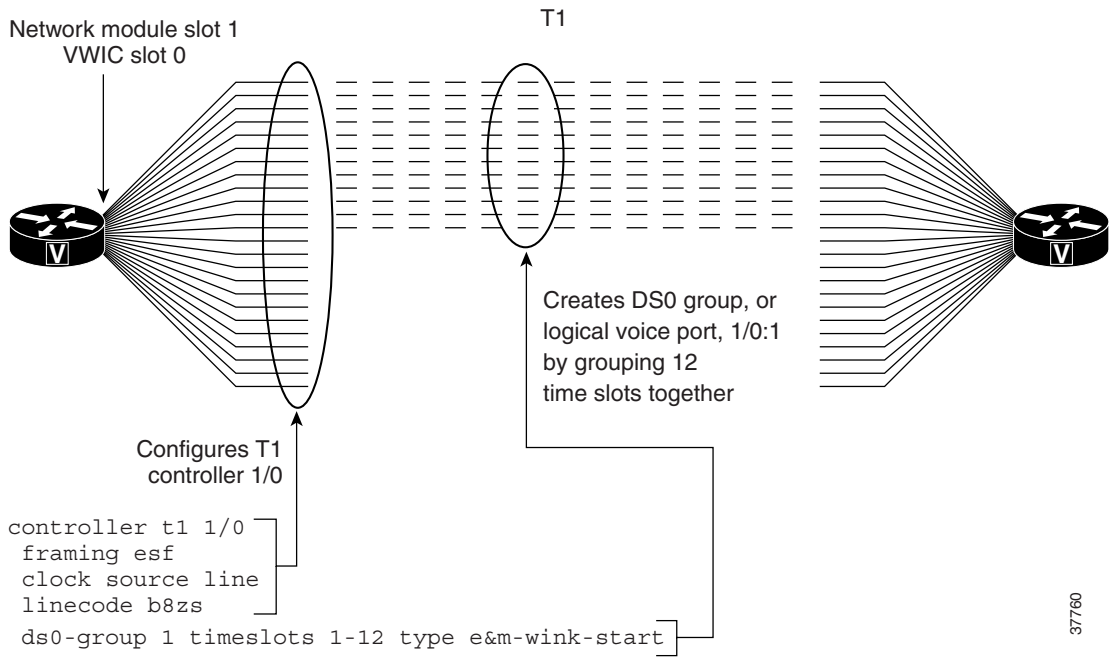
Configuring Controller Settings for Digital T1/E1 Voice Ports

The purpose of configuring controllers for digital T1/E1 voice ports is to match the configuration of the router to the line characteristics of the telephony network connection being made so that voice and signaling can be transferred between them and so that logical voice ports, or DS0 groups, may be established.

Figure 33 shows how a **ds0-group** command gathers some of the DS0 time slots from a T1 line into a group that becomes a single logical voice port, which can later be addressed as a single entity in voice port configurations. Other DS0 groups for voice can be created from the remaining time slots shown in the figure, or the time slots can be used for data or serial pass-through.

Note that all the controller commands in Figure 33 other than **ds0-group** apply to all the time slots in the T1.

Figure 33 T1 Controller Configuration on Cisco 2600 or 3600 Series Routers



Voice port controller configuration includes setting the parameters described in the following sections:

- Framing Formats on Digital T1/E1 Voice Ports
- Clock Sources on Digital T1/E1 Voice Ports
- Line Coding on Digital T1/E1 Voice Ports
- DS0 Groups on Digital T1/E1 Voice Ports

Another controller command that might be needed, **cablelength**, is discussed in the *Cisco IOS Interface Command Reference*, Release 12.2.

Framing Formats on Digital T1/E1 Voice Ports

The framing format parameter describes the way that bits are robbed from specific frames to be used for signaling purposes. The controller must be configured to use the same framing format as the line from the PBX or CO that connects to the voice port you are configuring.

Digital T1 lines use super frame (SF) or extended super frame (ESF) framing formats. SF provides two-state, continuous supervision signaling, in which bit values of 0 are used to represent on-hook and bit values of 1 are used to represent off-hook. ESF robs four bits instead of two, yet has little impact on voice quality. ESF is required for 64-kbps operation on DS0 and is recommended for Primary Rate Interface (PRI) configurations.

E1 lines can be configured for cyclic redundancy check (CRC4) or no cyclic redundancy check, with an optional argument for E1 lines in Australia.

Clock Sources on Digital T1/E1 Voice Ports

Digital T1/E1 interfaces use timers called *clocks* to ensure that voice packets are delivered and assembled properly. All interfaces handling the same packets must be configured to use the same source of timing so that packets are not lost or delivered late. The timing source that is configured can be external (from the line) or internal to the router's digital interface.

If the timing source is internal, timing derives from the onboard phase-lock loop (PLL) chip in the digital voice interface. If the timing source is line (external), then timing derives from the PBX or PSTN CO to which the voice port is connected. It is generally preferable to derive timing from the PSTN because their clocks are maintained at an extremely accurate level. This is the default setting for the clocks. When two or more controllers are configured, one should be designated as the primary clock source; it will drive the other controllers.

The **line** keyword specifies that the clock source is derived from the active line rather than from the free-running internal clock. The following rules apply to clock sourcing on the controller ports:

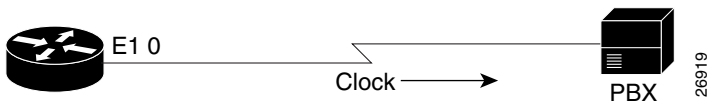
- When both ports are set to line clocking with no primary specification, port 0 is the default primary clock source and port 1 is the default secondary clock source.
- When both ports are set to line and one port is set as the primary clock source, the other port is by default the backup or secondary source and is loop-timed.
- If one port is set to clock source line or clock source line primary and the other is set to clock source internal, the internal port recovers clock from the clock source line port if the clock source line port is up. If it is down, then the internal port generates its own clock.
- If both ports are set to clock source internal, there is only one clock source: internal.

This section describes the five basic timing scenarios that can occur when a digital voice port is connected to a PBX or CO. In all the examples that follow, the PSTN (or CO) and the PBX are interchangeable for purposes of providing or receiving clocking.

- Single Voice Port Providing Clocking—In this scenario, the digital voice hardware is the clock source for the connected device, as shown in Figure 34. The PLL generates the clock internally and drives the clocking on the line. Generally, this method is useful only when connecting to a PBX, key system, or channel bank. A Cisco VoIP gateway rarely provides clocking *to* the CO because CO clocking is much more reliable. The following configuration sets up this clocking method for a digital E1 voice port:

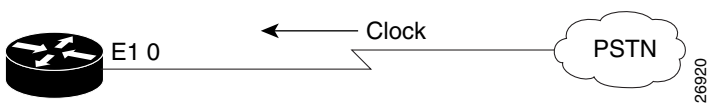
```
controller E1 1/0
 framing crc4
 linecoding hdb3
 clock source internal
 ds0-group timeslots 1-15 type e&m-wink-start
```

Figure 34 Single Voice Port Providing Clocking



- Single Voice Port Receiving Internal Clocking—In this scenario, the digital voice hardware receives clocking from the connected device (CO telephony switch or PBX) (see Figure 35). The PLL clocking is driven by the clock reference on the receive (Rx) side of the digital line connection.

Figure 35 Single E1 Port Receiving Clocking from the Line



The following configuration sets up this clocking method:

```
controller T1 1/0
 framing esf
 linecoding ami
 clock source line
 ds0-group timeslots 1-12 type e&m-wink-start
```

- Dual Voice Ports Receiving Clocking from the Line—In this scenario, the digital voice port has two reference clocks, one from the PBX and another from the CO, as shown in Figure 36. Because the PLL can derive clocking from only one source, this case is more complex than the two preceding examples.

  Before looking at the details, consider the following as they pertain to the clocking method:

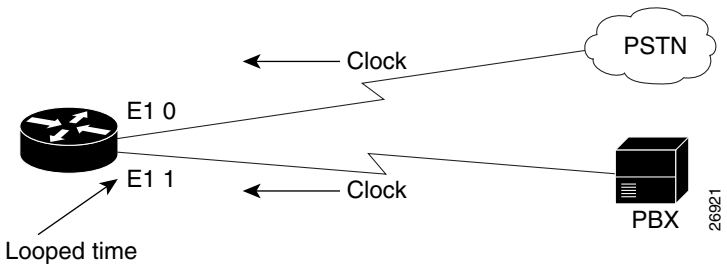
  – Looped-time clocking: The voice port takes the clock received on its Rx (receive) pair and regenerates it on its Tx (transmit) pair. While the port receives clocking, the port is not driving the PLL on the card but is "spoofing" (that is, fooling) the port so that the connected device has a viable clock and does not see slips (that is, loss of data bits). PBXs are not designed to accept slips on a T1 or E1 line, and such slips cause a PBX to drop the link into failure mode. While in looped-time mode, the router often sees slips, but because these are controlled slips, they usually do not force failures of the router's voice port.

  – Slips: These messages indicate that the voice port is receiving clock information that is out of phase (out of synchronization). Because the router has only a single PLL, it can experience controlled slips while it receives clocking from two different time sources. The router can usually handle controlled slips because its single-PLL architecture anticipates them.

Note Physical layer issues, such as bad cabling or faulty clocking references, can cause slips. Eliminate these slips by addressing the physical layer or clock reference problems.

In the dual voice ports receiving clocking from the line scenario, the PLL derives clocking from the CO and puts the voice port connected to the PBX into looped-time mode. This is usually the best method because the CO provides an excellent clock source (and the PLL usually requires that the CO provide that source) and a PBX usually must receive clocking from the other voice port.

Figure 36 Dual E1 Ports Receiving Clocking from the Line



The following configuration sets up this clocking method:

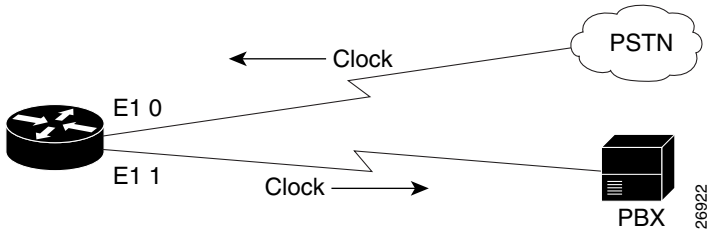
```
controller E1 1/0 << description - connected to the CO
 framing crc4
 linecoding hdb3
 clock source line primary
 ds0-group timeslots 1-15 type e&m-wink-start
!
```

```
controller E1 1/1 << description - connected to the PBX
 framing crc4
 linecoding hdb3
 clock source line
 ds0-group timeslots 1-15 type e&m-wink-start
```

The **clock source line primary** command tells the router to use this voice port to drive the PLL. All other voice ports configured as **clock source line** are then put into an implicit loop-timed mode. If the primary voice port fails or goes down, the other voice port instead receives the clock that drives the PLL. In this configuration, port 1/1 might see controlled slips, but these should not force it down. This method prevents the PBX from seeing slips.

- Dual Voice Ports (One Receives Clocking and One Provides Clocking)—In this scenario, the digital voice hardware receives clocking for the PLL from E1 0 and uses this clock as a reference to clock E1 1 (see Figure 37). If controller E1 0 fails, the PLL internally generates the clock reference to drive E1 1.

Figure 37 Dual E1 ports—One Receiving and One Providing Clocking

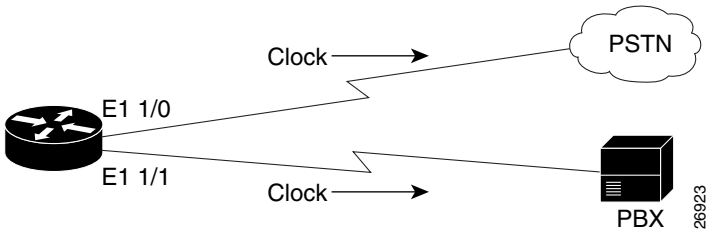


The following configuration sets up this clocking method:

```
controller E1 1/0
 framing crc4
 linecoding hdb3
 clock source line
 ds0-group timeslots 1-15 type e&m-wink-start
!
controller E1 1/1
 framing crc4
 linecoding hdb3
 clock source internal
 ds0-group timeslots 1-15 type e&m-wink-start
```

- Dual Voice Ports (Router Provides Both Clocks)—In this scenario, the router generates the clock for the PLL and, therefore, for both voice ports (see Figure 38).

Figure 38 Dual E1 Ports—both Clocks from the Router



The following configuration sets up this clocking method:

```
controller E1 1/0
 framing crc4
 linecoding hdb3
 clock source internal
 ds0-group timeslots 1-15 type e&m-wink-start
!
controller E1 1/1
 framing esf
 linecoding b8zs
 clock source internal
 ds0-group timeslots 1-15 type e&m-wink-start
```

Line Coding on Digital T1/E1 Voice Ports

Digital T1/E1 interfaces require that line encoding be configured to match that of the PBX or CO that is being connected to the voice port. Line encoding defines the type of framing used on the line.

T1 line encoding methods include alternate mark inversion (AMI) and binary 8 zero substitution (B8ZS). AMI is used on older T1 circuits and references signal transitions with a binary 1, or "mark." B8ZS, a more reliable method, is more popular and is recommended for PRI configurations as well. B8ZS encodes a sequence of eight zeros in a unique binary sequence to detect line-coding violations.

Supported E1 line encoding methods are AMI and high-density bipolar 3 (HDB3), which is a form of zero-suppression line coding.

DS0 Groups on Digital T1/E1 Voice Ports

For digital voice ports, a single command, **ds0-group**, performs the following functions:

- Defines the T1/E1 channels for compressed voice calls.
- Automatically creates a logical voice port.

  The numbering for the logical voice port created as a result of this command is *controller:ds0-group-no*, where *controller* is defined as the platform-specific address for a particular controller. On a Cisco 3640 router, for example, **ds0-group 1 timeslots 1-24 type e&m-wink** automatically creates the voice port 1/0:1 when issued in the configuration mode for controller 1/0. On a Cisco MC3810 universal concentrator, when you are in the configuration mode for controller 0, the command **ds0-group 1 timeslots 1-24 type e&m-wink** creates logical voice port 0:1.

  To map individual DS0s, define additional DS0 groups under the T1/E1 controller, specifying different time slots. Defining additional DS0 groups also creates individual DS0 voice ports.
- Defines the emulated analog signaling method that the router uses to connect to the PBX or PSTN.

  Most digital T1/E1 connections used for switch-to-switch (or switch-to-router) trunks are E&M connections, but FXS and FXO connections are also supported. These are normally used to provide emulated-OPX (Off-Premises eXtension) from a PBX to remote stations. FXO ports connect to FXS ports. The FXO or FXS connection between the router and switch (CO or PBX) must use matching signaling, or calls cannot connect properly. Either ground start or loop start signaling is appropriate for these connections. Ground start provides better disconnect supervision to detect when a remote user has hung up the telephone, but ground start is not available on all PBXs.

  Digital ground start differs from digital E&M because the A and B bits do not track each other as they do in digital E&M signaling (that is, A is not necessarily equal to B). When the CO delivers a call, it *seizes* a channel (goes off-hook) by setting the A bit to 0. The CO equipment also simulates ringing by toggling the B bit. The terminating equipment goes off-hook when it is ready to answer the call. Digits are usually not delivered for incoming calls.

E&M connections can use one of three different signaling types to acknowledge on-hook and off-hook states: wink start, immediate start, and delay start. E&M wink start is usually preferred, but not all COs and PBXs can handle wink start signaling. The E&M connection between the router and switch (CO or PBX) must match the CO or PBX E&M signaling type, or calls cannot be connected properly.

E&M signaling is normally used for trunks. It is normally the only way that a CO switch can provide two-way dialing with Direct Inward Dialing (DID). In all the E&M protocols, off-hook is indicated by A=B=1 and on-hook is indicated by A=B=0 (robbed-bit signaling). If dial pulse dialing is used, the A and B bits are pulsed to indicate the addressing digits. The are several further important subclasses of E&M robbed-bit signaling:

- E&M Wink Start—Feature Group B

  In the original wink start handshaking protocol, the terminating side responds to an off-hook from the originating side with a short wink (transition from on-hook to off-hook and back again). This wink tells the originating side that the terminating side is ready to receive addressing digits. After receiving addressing digits, the terminating side then goes off-hook for the duration of the call. The originating endpoint maintains off-hook for the duration of the call.

- E&M Wink Start—Feature Group D

  In Feature Group D wink start with wink acknowledge handshaking protocol, the terminating side responds to an off-hook from the originating side with a short wink (transition from on-hook to off-hook and back again) just as in the original wink start. This wink tells the originating side that the terminating side is ready to receive addressing digits. After receiving addressing digits, the terminating side provides another wink (called an *acknowledgment wink*) that tells the originating side that the terminating side has received the dialed digits. The terminating side then goes off-hook to indicate connection. This last indication can be due to the ultimate called endpoint's having answered. The originating endpoint maintains an off-hook condition for the duration of the call.

- E&M Immediate Start

  In the immediate-start protocol, the originating side does not wait for a wink before sending addressing information. After receiving addressing digits, the terminating side then goes off-hook for the duration of the call. The originating endpoint maintains off-hook for the duration of the call.

Note Feature Group D is supported on Cisco AS5300 platforms, and on Cisco 2600, 3600, and 7200 series with digital T1 packet voice trunk network modules. Feature Group D is not supported on E1 or analog voice ports.

To configure controller settings for digital T1/E1 voice ports, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**Cisco 7200 and 7500 series** `Router(config)# card type {t1 | e1} slot`	Defines the card as T1 or E1 and stipulates the location. The keywords and arguments are as follows: • **t1 | e1**—Defines the type of card. • *slot*—A value from 0 to 5.

Step 2	**Cisco 2600 and 3600 series, Cisco MC3810, and Cisco 7200 series**
	`Router(config)# controller {t1 | e1} slot/port`
	Cisco AS5300
	`Router(config)# controller {t1 | e1} number`
	Cisco AS5800
	`Router(config)# controller {t1 | e1} shelf/slot/port`
	Cisco 7500 series
	`Router(config)# controller {t1 | e1} slot/port-adapter/slot`
		Enters controller configuration mode.
		The keywords and arguments are as follows:
		• **t1 | e1**—The type of controller.
		• *slot/port*—The backplane slot number and port number for the interface being configured.
		• *number*—The network processor module number; the range is from 0 to 2.
		• *shelf/slot/port*—Indicates the controller ports; the range for *port* is from 0 to 11.

Step 3	**T1**
	`Router(config-controller)# framing {sf | esf}`
	E1
	`Router(config-controller)# framing {crc4 | no-crc4} [australia]`
		Selects frame type for T1 or E1 line.
		The keywords and arguments are as follows:
		T1 lines
		• **sf**—super frame
		• **esf**—extended super frame
		E1 lines
		• **crc4**—Provides 4 bits of error protection.
		• **no-crc4**—Disables **crc4**.
		• **australia**—(Optional) Specifies the E1 frame type used in Australia.
		The default for T1 is **sf**.
		The default for E1 is **crc4**.

	Command	Purpose
Step 4	`Router(config-controller)# clock source {line [primary | secondary] | internal}`	Configures the clock source. The keywords and arguments are as follows: • **line**—Specifies that the PLL on this port derives clocking from the external source to which the port is connected (generally the CO). • **primary**—(Optional) Specifies that the PLL on this port derives clocking from the external source and puts the other port (generally connected to the PBX) into looped-time mode. Both ports are configured with **line**, but only the port connected to the external source is configured with **primary**. • **secondary**—(Optional) Indicates a backup external source for clocking if the primary clocking shuts down. Configure the **clock source line secondary** command on the controller that has the next-best-known clocking. • **internal**—(Optional) Specifies that the clock is generated from the voice port's internal PLL. For more information about clock sources, see the "Clock Sources on Digital T1/E1 Voice Ports" section on page 102. The default is **line**.

Step 5	**T1 lines**
	`Router(config-controller)# linecode {ami | b8zs}`
	E1 lines
	`Router(config-controller)# linecode {ami | hdb3}`
		Specifies the line encoding to use.
		The keywords are as follows:
		• **ami**—Specifies the alternate mark inversion (AMI) line code type. (T1 and E1)
		• **b8zs**—Specifies the binary 8 zero substitution (B8ZS) line code type. (T1 only)
		• **hdb3**—Specifies the high-density bipolar 3 (HDB3) line code type. (E1 only)
		The default for T1 is **ami**.
		The default for E1 is **hdb3**.

	Command	Purpose
Step 6	**Cisco 2600 and 3600 Series Routers and Cisco MC3810 Multiservice Concentrators—T1** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {e&m-delay-dial | e&m-fgd | e&m-immediate-start e&m-wink-start | ext-sig | fgd-eana | fxo-ground-start | fxo-loop-start | fxs-ground-start | fxs-loop-start}` **Cisco 2600 and 3600 Series Routers and Cisco MC3810 Multservice Concentrators—E1** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {e&m-delay-dial | e&m-immediate-start | e&m-melcas-delay | e&m-melcas-immed | e&m-melcas-wink | e&m-wink-start | ext-sig | fgd-eana | fxo-ground-start | fxo-loop-start | fxo-melcas | fxs-ground-start | fxs-loop-start | fxs-melcas | r2-analog | r2-digital | r2-pulse}` **Cisco AS5300 Universal Access Servers—T1** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list [service {data | fax | voice}] [type {e&m-fgb | e&m-fgd | e&m-immediate-start | fxs-ground-start | fxs-loop-start | fgd-eana | fgd-os | r1-itu | sas-ground-start | sas-loop-start | none}]` **Cisco AS5300 Universal Access Servers—E1** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {none | p7 | r2-analog | r2-digital | r2-lsv181-digital | r2-pulse}` **Cisco AS5800 Universal Access Servers—T1** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {e&m-fgb | e&m-fgd | e&m-immediate-start | fxs-ground-start | fxs-loop-start | fgd-eana | r1-itu | r1-modified | r1-turkey | sas-ground-start | sas-loop-start | none}` **Cisco AS5800 Universal Access Servers E1 Voice Ports** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {e&m-fgb | e&m-fgd | e&m-immediate-start | fxs-ground-start | fxs-loop-start | p7 | r2-analog | r2-digital | r2-pulse | sas-ground-start | sas-loop-start | none}` **Cisco 7200 and 7500 Series Series Routers T1 and E1 Voice Ports** `Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {e&m-delay | e&m-immediate | e&m-wink | fxs-ground-start | fxs-loop-start | fxo-ground-start | fxo-loop-start}`	Defines the T1 channels for use by compressed voice calls and the signaling method that the router uses to connect to the PBX or CO. **Note** This step shows the basic syntax and signaling types available with the **ds0-group** command. For the complete syntax, see the *Cisco IOS Voice, Video, and Fax Command Reference*, Release 12.2. The keywords and arguments are as follows: • *ds0-group-no*—Identifies the DS0 group (number from 0 to 23, for T1, or from 0 to 30, for E1). • **timeslots** *timeslot-list*—Specifies the single time slot number, single range of numbers, or multiple ranges of numbers separated by commas. For T1/E1, allowable values are from 1 to 24. Examples are as follows: – 2, 3-5 – 1, 7, 9 – 1-12 • **service—**Indicates the type of calls to be handled by this DS0 group—data, fax, or voice). • **type—**Refers to the signaling type of the telephony connection being made. Types include the following: – **e&m-delay-dial**—Specifies the originating endpoint that sends an off-hook signal and waits for the off-hook signal followed by an on-hook signal from the destination. – **e&m-fgb**—E & M Type II Feature Group B. – **e&m-fgd**—E & M Type II Feature Group D.

Command	Purpose
	– **e&m-immediate-start**—E & M Immediate Start. – **e&m-melcas-delay**—E&M Mercury Exchange Limited Channel Associated Signaling (MELCAS) delay start signaling support. – **e&m-melcas-immed**—E&M MELCAS immediate start signaling support. – **e&m-melcas-wink**—E&M MELCAS wink start signaling support. – **e&m-wink-start**—The originating endpoint sends an off-hook signal and waits for a – **ext-sig**—For the specified channel, automatically generates the off-hook signal and stays in the off-hook state. – **fgd-eana**—Feature Group D Exchange Access North American. – **fgd-os**—Feature Group D Operator Services. – **fxo-melcas**—MELCAS Foreign Exchange Office signaling support. – **fxs-melcas**—MELCAS Foreign Exchange Station signaling support. – **fxs-ground-start**—FXS Ground Start. – **fxs-loop-start**—FXS Loop Start. – **none**—Null Signaling for External Call Control. – **p7**—Specifies the p7 switch type. – **r1-itu**—R1 ITU – **sas-ground-start**—SAS Ground Start. – **sas-loop-start**—SAS Loop Start. The **r1** and **r2** keywords refer to line signaling, based on international signaling standards. The **r1 itu** keywords are based on signaling standards in countries besides the United States. An "ITU variant" means that there are multiple R1 standards in a particular country but that Cisco supports the ITU variant.
Step 7 `Router(config-controller)# no shutdown`	Activates the controller.

Configuring Basic Voice Port Parameters for Digital T1/E1 Voice Ports

For FXO and FXS connections the default voice-port parameter values are often adequate. However, for E&M connections, it is important to match the characteristics of your PBX, so voice port parameters may need to be reconfigured from their defaults.

Each voice port that you address in digital voice port configuration is one of the logical voice ports that you created with the **ds0-group** command.

Companding (from *compression* and *expansion*), used in Step 4 of the following table, is the part of the PCM process in which analog signal values are logically rounded to discrete scale-step values on a nonlinear scale. The decimal step number is then coded in its binary equivalent prior to transmission. The process is reversed at the receiving terminal using the same nonlinear scale.

Note The commands, keywords, and arguments that you are able to use may differ slightly from those presented here, based on your platform, Cisco IOS release, and configuration. When in doubt, use Cisco IOS command help (**command ?**) to determine the syntax choices that are available.

To configure basic parameters for digital T1/E1 voice ports, use the following commands beginning in global configuration mode.

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers**
	`Router(config)# voice-port slot/port:ds0-group-no`
	Cisco MC3810 Multiseries Concentrators
	`Router(config)# voice-port slot:ds0-group-no`
	Cisco AS5300 Universal Access Server
	`Router(config)# voice-port controller:ds0-group-no`
	Cisco AS5800 Universal Access Server
	`Router(config)# voice-port shelf/slot/port:ds0-group-no`
	Cisco 7200 Series Routers
	`Router(config)# voice-port slot/port-adapter:ds0-group-no`
	Cisco 7500 Series Routers
	`Router(config)# voice-port slot/port-adapter/slot:ds0-group-no`
		Enters voice-port configuration mode. The arguments are defined as the following
		• *slot*—Specifies the router location where the network module (Cisco 2600, 3600, and MC3810) or voice port adapter (Cisco AS5300, AS5800, 7200, and 7500) is installed. This is the same number as the controller for the T1/E1 voice port.
		• *port*—Indicates the voice interface card location.
		• *ds0-group-no*—Specifies the logical voice port that was created with the **ds0-group** controller command.
		• *controller*—Indicates the controller for the T1/E1 voice port.
		• *shelf*—Specifies the dial shelf, which is always 0.
		• *port-adapter*—Indicates the port adapter for the voice port.

Step 2	`Router(config-voiceport)# type {1 | 2 | 3 | 5}`	(E&M only) Specifies the type of E&M interface to which this voice port is connected. See Table 8 for an explanation of E&M types. The default is 1.

	Command	Purpose
Step 3	`Router(config-voiceport)# cptone` *locale*	Selects a two-letter **locale** keyword for the voice call progress tones and other locale-specific parameters to be used on this voice port. Voice call progress tones include dial tone, busy tone, and ringback tone, which vary with geographical region. Other parameters include ring cadence and compand type. Cisco routers comply with the ISO3166 locale name standards; to see valid choices, enter a question mark (?) following the **cptone** command. The default is **us**.
Step 4	`Router(config-voiceport)# compand-type {u-law | a-law}`	(Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrators only) Specifies the companding standard used. This command is used in cases when the DSP is not used, such as local cross-connects, and overwrites the **compand-type** value set by the **cptone** command. The keywords are as follows: • **a-law**—Specifies the ITU-T PCM a-law companding standard used primarily in Europe. The default for E1 is **a-law**. • **u-law**—Specifies the ITU-T PCM mu-law companding standard used in North America and Japan. The default for T1 is **u-law**. **Note** If you have a Cisco MC3810 multiservice concentrator or Cisco 3660 router, the **compand-type a-law** command must be configured on the analog ports only. The Cisco 2660, 3620, and 3640 routers do not require the **compand-type a-law** command configured, however, if you request a list of commands, the **compand-type a-law** command will display.

Step 5	**Cisco 2600 series and 3600 series**
	`Router(config-voiceport)# ring frequency {25 | 50}`
	
	Cisco MC3810
	`Router(config-voiceport)# ring frequency {20 | 30}`
		(FXS only) Selects the ring frequency, in hertz, used on the FXS interface. This number must match the connected telephony equipment, and can be country-dependent. If not set properly, the attached telephony device may not ring or it may buzz.
		
		The default is 25 on the Cisco 2600 and 3600 series routers and 20 on the Cisco MC3810 multiservice concentrators.

	Command	Purpose
Step 6	`Router(config-voiceport)# ring number number`	(FXO only) Specifies the maximum number of rings to be detected before an incoming call is answered by the router. The default is 1.
Step 7	`Router(config-voiceport)# ring cadence {[pattern01 | pattern02 | pattern03 | pattern04 | pattern05 | pattern06 | pattern07 | pattern08 | pattern09 | pattern10 | pattern11 | pattern12] [define pulse interval]}`	(FXS only) Specifies an existing pattern for ring, or defines a new one. Each pattern specifies a ring-pulse time and a ring-interval time. The keywords and arguments are as follows: • **pattern01** through **pattern12**—Specifies preset ring cadence patterns. Enter **ring cadence ?** to see ring pattern explanations. • **define** *pulse interval*—Specifies a user-defined pattern as follows: – *pulse* is a number (1 or 2 digits from 1 to 50) specifying ring pulse (on) time in hundreds of milliseconds. – *interval* is a number (1 or 2 digits from 1 to 50) specifying ring interval (off) time in hundreds of milliseconds. The default is the pattern specified by the configured **cptone locale** command.
Step 8	`Router(config-voiceport)# description string`	Attaches a text string to the configuration that describes the connection for this voice port. This description appears in various displays and is useful for tracking the purpose or use of the voice port. The *string* argument is a character string from 1 to 255 characters in length. The default is that no description is attached to the configuration.
Step 9	`Router(config-voiceport)# no shutdown`	Activates the voice port.

Fine-Tuning Analog and Digital Voice Ports

Normally, default parameter values for voice ports are sufficient for most networks. Depending on the specifics of your particular network, however, you may need to adjust certain parameters that are configured on voice ports. Collectively, these commands are referred to as voice port tuning commands.

Note The commands, keywords, and arguments that you are able to use may differ slightly from those presented here, based on your platform, Cisco IOS release, and configuration. When in doubt, use Cisco IOS command help (**command ?**) to determine the syntax choices that are available.

The voice port tuning commands are grouped into these categories and explained in the following sections:

Full descriptions of the commands in this section can be found in the *Cisco IOS Voice, Video, and Fax Command Reference*, Release 12.2.

Auto Cut-Through Command

The **auto-cut-through** command allows you to connect to PBXs that do not provide an M-lead response.

To configure auto-cut-through, use the following command in voice-port configuration mode:

Command	Purpose
`Router(config-voiceport)# auto-cut-through`	(E&M only) Enables call completion on a router when a PBX does not provide an M-lead response.

Bit Modification Commands for Digital Voice Ports

The bit modification commands for digital voice ports modify sent or received bit patterns. Different versions of E&M use different ABCD signaling bits to represent idle and seize. For example, North American CAS E&M represents idle as 0XXX and seize as 1XXX, where X indicates that the state of the BCD bits is ignored. In MELCAS E&M, idle is 1101 and seize is 0101. The commands in this section are provided to modify bit patterns to match particular E&M schemes.

To manipulate bit patterns for digital voice ports, use the following commands as necessary, in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# condition {tx-a-bit | tx-b-bit | tx-c-bit | tx-d-bit} {rx-a-bit | rx-b-bit | rx-c-bit | rx-d-bit} {on | off | invert}`	Manipulates sent or received bit patterns to match expected patterns on a connected device. Repeat the command for each transmit and/or receive bit to be modified, but be careful not to destroy the information content of the bit pattern. The default is that the signaling format is not manipulated (for all transmit or receive A, B, C, and D bits).
		The keywords are as follows: • **on**—Sets the bit to 1 permanently. • **off**—Sets the bit to 0 permanently. • **invert**—Changes the state to the opposite of the original transmit or receive state. **Note** The **show voice port** command reports at the protocol level, and the **show controller** command reports at the driver level. The driver is not notified of any bit manipulation using the **condition** command. As a result, the **show controller** command output does not account for the bit conditioning.
Step 2	`Router(config-voiceport)# define {tx-bits | rx-bits} {seize | idle} {0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111}`	(Digital E1 E&M voice ports on Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrators only) Defines specific transmit or receive signaling bits to match the bit patterns required by a connected device for North American E&M and E&M MELCAS voice signaling, if patterns different from the preset defaults are required. Also specifies which bits a voice port monitors and which bits it ignores, if patterns that are different from the defaults are required. See the **define** command for the default signaling patterns as defined in American National Standards Institute (ANSI) and code excited linear prediction compression (CEPT) standards. The keywords are as follows: • **tx-bits**—Indicates the pattern applies to transmit signaling bits.

	Command	Purpose
		• **rx-bits**—Indicates the pattern applies to receive signaling bits • **seize**—Indicates that the pattern represents line seizure. • **idle**—Indicates that the pattern represents an idle condition. • **0000**...**1111**—Represents the bit pattern to use.
Step 3	`Router(config-voiceport)# ignore {rx-a-bit | rx-b-bit | rx-c-bit | rx-d-bit}`	(Digital E1 E&M voice ports on Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrators only) Configures the voice port to ignore the specified receive bit for North American E&M or E&M MELCAS, if patterns different from the defaults are required. See the command reference for the default signaling patterns as defined in ANSI and CEPT standards.

Calling Number Outbound Commands

On the Cisco AS5300 universal access server platform, if T1 CAS is configured with the Feature Group-D (FGD)—Exchange Access North American (FGD-EANA) signaling, the automatic number identification (ANI) can be sent for outgoing calls by using the **calling-number outbound** command.

FGD-EANA is a FGD signaling protocol of type EANA, which provides certain call services, such as emergency (USA 911) calls. ANI is an SS7 (Signaling System 7) feature in which a series of digits, analog or digital, are included in the call to identify the telephone number of the calling device. In other words, ANI identifies the number of the calling party. ANI digits are used for billing purposes by Internet service providers (ISPs), among other things. The commands in this section can be issued in voice-port or dial-peer mode, because the syntax is the same.

To configure your digital T1/E1 packet voice trunk network module to generate outbound ANI digits on a Cisco AS5300 universal access server, use the following commands in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# calling-number outbound range string1 string2`	(Cisco AS5300 universal access server only) Specifies ANI to be sent out when the T1-CAS **fgd-eana** command is configured as signaling type. The *string1* and *string2* arguments are valid E.164 telephone number strings. Both strings must be of the same length and cannot be more than 32 digits long. Only the last four digits are used for specifying the range (*string1* to *string2*) and for generating the sequence of ANI by rotating through the range until *string2* is reached and then starting from *string1* again. If strings are less than four digits in length, then entire strings are used.
Step 2	`Router(config-voiceport)# calling-number outbound sequence [string1] [string2] [string3] [string4] [string5]`	(Cisco AS5300 universal access server only) Specifies ANI to be sent out when the T1-CAS **fgd-eana** command is configured as signaling type. This option configures a sequence of discrete strings (*string1...string5*) to be passed out as ANI for successive calls using the dial peer or voice port. Limit is five (5) strings. All strings must be valid E.164 numbers, up to 32 digits in length.
Step 3	`Router(config-voiceport)# calling-number outbound null`	(Cisco AS5300 universal access server only) Suppresses ANI. No ANI is passed when this voice port is selected.

Disconnect Supervision Commands

PBX and PSTN switches use several different methods to indicate that a call should be disconnected because one or both parties have hung up. The commands in this section are used to configure the router to recognize the type of signaling in use by the PBX or PSTN switch connected to the voice port. These methods include the following:

- Battery reversal disconnect
- Battery denial disconnect
- Supervisory tone disconnect (STD)

Battery reversal occurs when the connected switch changes the polarity of the line in order to indicate changes in call state (such as off-hook or, in this case, call disconnect). This is the signaling looked for when the **battery reversal** command is enabled on the voice port, which is the default configuration.

Battery denial (sometimes called *power denial*) occurs when the connected switch provides a short (approximately 600 ms) interruption of line power to indicate a change in call state. This is the signaling looked for when the **supervisory disconnect** command is enabled on the voice port, which is the default configuration.

Supervisory tone disconnect occurs when the connected switch provides a special tone to indicate a change in call state. Some PBXs and PSTN CO switches provide a 600-millisecond interruption of line power as a supervisory disconnect, and others provide supervisory tone disconnect (STD). This is the signal that the router is looking for when the **no supervisory disconnect** command is configured on the voice port.

Note In some circumstances, you can use the FXO Disconnect Supervision feature to enable analog FXO ports to monitor call progress tones for disconnect supervision that are returned from a PBX or from the PSTN. For more information, see the "FXO Supervisory Disconnect Tone Commands" section on page 121.

To change parameters related to disconnect supervision, use the following commands as appropriate, in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# no battery-reversal`	(Analog only) Enables battery reversal. The default is that battery reversal is enabled. • For FXO ports—Use the **no battery-reversal** command to configure a loop-start voice port not to disconnect when it detects a second battery reversal. The default is to disconnect when a second battery reversal is detected. This functionality is supported on Cisco MC3810 analog voice ports; on Cisco 1750, Cisco 2600 series, and Cisco 3600 series routers, only analog voice ports on VIC-2FXO cards are able to detect battery reversal. – Also use the **no battery-reversal** command when a connected FXO port does not support battery reversal detection. • For FXS ports—Use the **no battery-reversal** command to configure the voice port not to reverse battery when it connects calls. The default is to reverse battery when a call is connected, then return to normal when the call is over, providing positive disconnect. See also the **disconnect-ack** command (Step 7).
Step 2	`Router(config-voiceport)# no supervisory disconnect`	(FXO only) Enables the PBX or PSTN switch to provide STD. By default the **supervisory disconnect** command is enabled.
Step 3	`Router(config-voiceport)# disconnect-ack`	(FXS only) Configures the voice port to return an acknowledgment upon receipt of a disconnect signal. The FXS port removes line power if the equipment on the FXS loop-start trunk disconnects first. This is the default. The **no disconnect-ack** command prevents the FXS port from responding to the on-hook disconnect with a removal of line power.

FXO Supervisory Disconnect Tone Commands

If the FXO supervisory disconnect tone is configured and a detectable tone from the PSTN or PBX is detected by the digital signal processor (DSP), the analog FXO port goes on-hook. This feature prevents an analog FXO port from remaining in an off-hook state after an incoming call is ended. FXO supervisory disconnect tone enables interoperability with PSTN and PBX systems whether or not they transmit supervisory tones.

Note This feature applies only to analog FXO ports with loop-start signaling on the Cisco 2600 and 3600 series routers and on Cisco MC3810 multiservice concentrators with high-performance compression modules (HCMs).

To configure a voice port to detect incoming tones, you need to know the parameters of the tones expected from the PBX or PSTN. Then create a voice class that defines the tone detection parameters, and, finally, apply the voice class to the applicable analog FXO voice ports. This procedure configures the voice port to go on-hook when it detects the specified tones. The parameters of the tones need to be precisely specified to prevent unwanted disconnects due to detection of nonsupervisory tones or noise.

A supervisory disconnect tone is normally a dual tone consisting of two frequencies; however, tones of only one frequency can also be detected. Use caution if you configure voice ports to detect nondual tones, because unwanted disconnects can result from detection of random tone frequencies. You can configure a voice port to detect a tone with one on/off time cycle, or you can configure it to detect tones in a cadence pattern with up to four on/off time cycles.

To create a voice class that defines the specific tone or tones to be detected and then apply the voice class to the voice port, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class dualtone` *tag*	Creates a voice class for defining one tone detection pattern. The range for the tag number is from 1 to 10000. The tag number must be unique on the router. For more information about configuring voice classes, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide.
Step 2	`Router(config-voice-class)# freq-pair` *tone-id frequency-1 frequency-2*	Specifies the two frequencies, in Hz, for a tone to be detected (or one frequency if a nondual tone is to be detected). If the tone to be detected contains only one frequency, enter 0 for *frequency-2*. The arguments are as follows: • *tone-id*—Ranges from 1 to 16. There is no default. • *frequency-1* and *frequency-2*—Ranges from 300 to 3600, or you can enter 0 for *frequency-2*. There is no default. **Note** Repeat this command for each additional tone to be specified.

	Command	Purpose
Step 3	`Router(config-voice-class)#` **freq-max-deviation** *frequency*	Specifies the maximum frequency deviation that will be detected, in Hz. The *frequency* argument ranges from 10 to 125. The default is 10.
Step 4	`Router(config-voice-class)#` **freq-max-power** *dBmO*	Specifies the maximum tone power that will be detected, in dBmO. The *dBmO* argument ranges from 0 to 20. The default is 10.
Step 5	`Router(config-voice-class)#` **freq-min-power** *dBmO*	Specifies the minimum tone power that will be detected, in dBmO. The *dBmO* argument ranges from 10 to 35. The default is 30.
Step 6	`Router(config-voice-class)#` **freq-power-twist** *dBmO*	Specifies the power difference allowed between the two frequencies, in dBmO. The *dBmO* argument ranges from 0 to 15. The default is 6.
Step 7	`Router(config-voice-class)#` **freq-max-delay** *time*	Specifies the timing difference allowed between the two frequencies, in 10-millisecond increments. The *time* argument ranges from 10 to 100 (100 ms to 1 s). The default is 20 (200 ms).
Step 8	`Router(config-voice-class)#` **cadence-min-on-time** *time*	Specifies the minimum tone on time that will be detected, in 10-millisecond increments. The *time* argument ranges from 0 to 100 (0 ms to 1 s).
Step 9	`Router(config-voice-class)#` **cadence-max-off-time** *time*	Specifies the maximum tone off time that will be detected, in 10-millisecond increments. The *time* argument ranges from 0 to 5000 (0 ms to 50 s).
Step 10	`Router(config-voice-class)#` **cadence-list** *cadence-id cycle-1-on-time cycle-1-off-time cycle-2-on-time cycle-2-off-time cycle-3-on-time cycle-3-off-time cycle-4-on-time cycle-4-off-time*	(Optional) Specifies a tone cadence pattern to be detected. Specify an on time and off time for each cycle of the cadence pattern. The arguments are as follows: • *cadence-id*—Ranges from 1 to 10. There is no default. • *cycle-N-on-time* and *cycle-N-off-time*—Range from 0 to 1000 (0 ms to 10 s). The default is 0.
Step 11	`Router(config-voice-class)#` **cadence-variation** *time*	(Optional) Specifies the maximum time that the tone onset can vary from the specified onset time and still be detected, in 10-millisecond increments. The *time* argument ranges from 0 to 200 (0 ms to 2 s). The default is 0.
Step 12	`Router(config-voice-class)#` **exit**	Exits voice class configuration mode.

	Command	Purpose

Step 13	**Cisco 2600 and 3600 Series Routers**
	`Router(config)# voice-port slot/subunit/port`
	
	Cisco MC3810 Multiservice Concentrators
	`Router(config)# voice-port slot/port`
		Enters voice-port configuration mode.
		The arguments are as follows:
		• *slot*—Specifies the slot number where the voice network module is installed (Cisco 2600 and Cisco 3600 series routers) or the router slot number where the analog voice module is installed (Cisco MC3810 multiservice concentrators).
		• *subunit*—Specifies the voice interface card (VIC) where the voice port is located.
		• *port*—Identifies the analog voice-port number.

Step 14	`Router(config-voiceport)# supervisory disconnect dualtone {mid-call | pre-connect} voice-class tag`	Assigns an FXO supervisory disconnect tone voice class to the voice port. The keywords are as follows: • **mid-call**—Specifies tone detection during the entire call. • **pre-connect**—Specifies tone detection only during call set-up.
Step 15	`Router(config-voiceport)# supervisory disconnect anytone`	Configures the voice port to disconnect on receipt of any tone.

Timeouts Commands

To change timeouts parameters, use the following commands as appropriate, in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# timeouts call-disconnect seconds`	Configures the call disconnect timeout value in seconds. Valid entries range from 0 to 120. The default is 60.
Step 2	`Router(config-voiceport)# timeouts initial seconds`	Sets the number of seconds that the system waits between the caller input of the initial digit and the subsequent digit of the dialed string. If the wait time expires before the destination is identified, a tone sounds and the call ends. The *seconds* argument is the initial timeout duration. A valid entry is an integer from 0 to 120. The default is 10.

	Command	Purpose
Step 3	`Router(config-voiceport)# timeouts interdigit` *seconds*	Configures the number of seconds that the system waits after the caller has input the initial digit or a subsequent digit of the dialed string. If the timeout ends before the destination is identified, a tone sounds and the call ends. This value is important when using variable-length dial peer destination patterns (dial plans). The *seconds* argument is the interdigit timeout wait time in seconds. A valid entry is an integer from 0 to 120. The default is 10.
Step 4	`Router(config-voiceport)# timeouts ringing` {*seconds* | **infinity**}	Specifies the duration that the voice port allows ringing to continue if a call is not answered. The keyword and argument are as follows: • **infinity**—Indicates ringing should continue until the caller goes on hook. • *seconds*—Specifies the number of seconds to allow ringing without answer. The range is from 5 to 60000. The default is 180.
Step 5	`Router(config-voiceport)# timeouts wait-release` {*seconds* | **infinity**}	Specifies the duration that a voice port stays in the call-failure state while the Cisco device sends a busy tone, reorder tone, or an out-of-service tone to the port. The keyword and argument are as follows: • **infinity**—Indicates the voice port should not be released as long as the call-failure state remains. • *seconds*—Specifies the number of seconds to allow before the call is released. The range is from 3 to 3600. The default is 30.

Timing Commands

To change timing parameters, use the following commands as appropriate, in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# timing clear-wait` *milliseconds*	(E&M only) Specifies the minimum amount of time between the inactive seizure signal and clearing of the call. Valid entries for the *milliseconds* argument are from 200 to 2000 milliseconds. The default is 400.
Step 2	`Router(config-voiceport)# timing delay-duration` *milliseconds*	(E&M only) Specifies the delay signal duration for delay-dial signaling in milliseconds. Valid entries are from 100 to 5000. The default is 2000.
Step 3	`Router(config-voiceport)# timing delay-start` *milliseconds*	(E&M only) Specifies minimum delay time, in milliseconds, from outgoing seizure to outdial address. Valid entries are from 20 to 2000. The default is 300 for the Cisco 3600 series routers, and 150 for the Cisco MC3810 multiservice concentrators.
Step 4	`Router(config-voiceport)# timing delay-with-integrity` *milliseconds*	(Cisco MC3810 multiservice concentrators E&M ports only) Specifies duration of the wink pulse for the delay dial in milliseconds. Valid entries are from 0 to 5000. The default is 0.
Step 5	`Router(config-voiceport)# timing dial-pulse min-delay` *milliseconds*	Specifies time, in milliseconds, between the generation of wink-like pulses when the type is pulse. Valid entries are from 0 to 5000. The default is 300 for the Cisco 3600 series routers, and 140 for the Cisco MC3810 multiservice concentrators.
Step 6	`Router(config-voiceport)# timing dialout-delay` *milliseconds*	(Cisco MC3810 multiservice concentrators only) Specifies dialout delay, in milliseconds, for the sending digit or cut-through on an FXO trunk or an E&M immediate trunk. Valid entries are from 100 to 5000. The default is 300.
Step 7	`Router(config-voiceport)# timing digit` *milliseconds*	Specifies the DTMF digit signal duration in milliseconds. Valid entries are from 50 to 100. The default is 100.
Step 8	`Router(config-voiceport)# timing guard-out` *milliseconds*	(FXO ports only) Specifies the duration in milliseconds of the guard-out period that prevents this port from seizing a remote FXS port before the remote port detects a disconnect signal. The range is from 300 to 3000. The default is 2000.
Step 9	`Router(config-voiceport)# timing hookflash-out` *milliseconds*	Specifies the duration, in milliseconds, of the hookflash. Valid entries are from 50 to 500. The default is 300.

	Command	Purpose
Step 10	`Router(config-voiceport)#` **`timing interdigit`** *`milliseconds`*	Specifies the DTMF interdigit duration, in milliseconds. Valid entries are from 50 to 500. The default is 100.
Step 11	`Router(config-voiceport)#` **`timing percentbreak`** *`percent`*	(Cisco MC3810 multiservice concentrators FXO and E&M ports only) Specifies the percentage of the break period for the dialing pulses, if different from the default. The range is from 20 to 80. The default is 50**.**
Step 12	`Router(config-voiceport)#` **`timing pulse`** *`pulses-per-second`*	(FXO and E&M only) Specifies the pulse dialing rate in pulses per second. Valid entries are from 10 to 20. The default is 20.
Step 13	`Router(config-voiceport)#` **`timing pulse-digit`** *`milliseconds`*	(FXO only) Configures the pulse digit signal duration. The range of the pulse digit signal duration is from 10 to 20. The default is 20.
Step 14	`Router(config-voiceport)#` **`timing pulse-interdigit`**	(FXO and E&M only) Specifies pulse dialing interdigit timing in milliseconds. Valid entries are from 100 to 1000. The default is 500.
Step 15	`Router(config-voiceport)#` **`timing wink-duration`** *`milliseconds`*	(E&M only) Specifies maximum wink-signal duration, in milliseconds, for a wink-start signal. Valid entries are from 100 to 400. The default is 200.
Step 16	`Router(config-voiceport)#` **`timing wink-wait`** *`milliseconds`*	(E&M only) Specifies maximum wink-wait duration, in milliseconds, for a wink-start signal. Valid entries are from 100 to 5000. The default is 200.

DTMF Timer Inter-Digit Command for Cisco AS5300 Access Servers

To configure the DTMF timer for Cisco AS5300 access servers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)#` **`controller T1`** *`number`*	Configures a T1 controller and enters controller configuration mode.
Step 2	`Router(config)#` **`ds0-group`** *`channel-number`* **`timeslots`** *`range`* **`type`** *`signaling-type`* **`dtmf dnis`**	Configures channelized T1 timeslots, which enables a Cisco AS5300 modem to answer and send an analog call.
Step 3	`Router(config)#` **`cas-custom`** *`channel`*	Customizes E1 R2 signaling parameters for a particular E1 channel group on a channelized E1 line.
Step 4	`Router(conf-ctrl-cas)#` **`dtmf-timer-inter-digit`** *`milliseconds`*	Configures the DTMF inter-digit timer for a DS0 group.

Verifying DTMF Timer Inter-Digit Command

To verify the DTMF timer, use the following command in EXEC mode:

Command	Purpose
`Router# show running-config`	Displays the configuration information currently running on the terminal.

Voice Activity Detection Commands Related to Voice-Port Configuration Mode

In normal voice conversations, only one person speaks at a time. Today's circuit-switched telephone networks dedicate a bidirectional, 64 kbps channel for the duration of each conversation, regardless of whether anyone is speaking at the moment. This means that, in a normal voice conversation, at least 50 percent of the bandwidth is wasted when one or both parties are silent. This figure can actually be much higher when normal pauses and breaks in conversation are taken into account.

Packet-switched voice networks, on the other hand, can use this "wasted" bandwidth for other purposes when voice activity detection (VAD) is configured. VAD works by detecting the magnitude of speech in decibels and deciding when to cut off the voice from being framed. VAD has some technological problems, however, which include the following:

- General difficulties determining when speech ends
- Clipped speech when VAD is slow to detect that speech is beginning again
- Automatic disabling of VAD when conversations take place in noisy surroundings

VAD is configured on dial peers; by default it is enabled. For more information, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide. Two parameters associated with VAD, music threshold and comfort noise, are configured on voice ports.

If VAD is enabled, use the following commands to adjust parameter values associated with VAD, beginning in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# music-threshold` *number*	Specifies the minimal decibel level of music played when calls are put on hold. The decibel level affects how voice activity detection (VAD) treats the music data. Valid entries range from –70 to –30. When used with VAD, if the level is set too high, the remote end hears no music; if it is set too low, there is unnecessary voice traffic. The default is –38.
Step 2	`Router(config-voiceport)# comfort-noise`	This parameter creates subtle background noise to fill silent gaps during calls when VAD is enabled on voice dial peers. If comfort noise is not generated, the resulting silence can fool the caller into thinking the call is disconnected instead of being merely idle. The default is that comfort noise is enabled.

Voice Quality Tuning Commands

The commands in this section configure parameters to improve voice quality. Common voice quality issues include the following:

Delay in Voice Networks

Delay is the time it takes for voice packets to travel between two endpoints. Excessive delay can cause quality problems with real-time traffic such as voice. However, because of the speed of network links and the processing power of intermediate devices, some delay is expected.

When listening to speech, the human ear normally accepts up to about 150 ms of delay without noticing delays. The ITU G.114 standard recommends no more than 150 ms of one-way delay for a normal voice conversation. Once the delay exceeds 150 ms, a conversation is more like a “walkie-talkie” conversation in which one person must wait for the other to stop speaking before beginning to talk.

You can measure delay fairly easily by using ping tests at various times of the day with different network traffic loads. If network delay is excessive, it must be reduced for adequate voice quality.

Several different types of delay combine to make up the total end-to-end delay associated with voice calls:

- Propagation delay—Amount of time it takes the data to physically travel over the media.
- Handling delay—Amount of time it takes to process data by adding headers, taking samples, forming packets, etc.
- Queuing delay—Amount of time lost due to congestion.
- Variable delay or jitter—Amount of time that causes the conversation to break and become unintelligible. Jitter is described in detail below.

Propagation, handling, and queuing delay are not addressed by voice-port commands and fall outside the scope of this chapter.

Jitter Adjustment

Delay can cause unnatural starting and stopping of conversations, but variable-length delays (also known as jitter) can cause a conversation to break and become unintelligible. Jitter is not usually a problem with PSTN calls because the bandwidth of calls is fixed and each call has a dedicated circuit for the duration of the call. However, in VoIP networks, data traffic might be bursty, and jitter from the packet network can become an issue. Especially during times of network congestion, packets from the same conversation can arrive at different interpacket intervals, disrupting the steady, even delivery needed for voice calls. Cisco voice gateways have built-in jitter buffering to compensate for a certain amount of jitter; the **playout-delay** command can be used to adjust the jitter buffer.

Normally, the defaults in effect are sufficient for most networks. However, a small playout delay from the jitter buffer can cause lost packets and choppy audio, and a large playout delay can cause unacceptably high overall end-to-end delay.

Note Prior to Cisco IOS Release 12.1(5)T, playout delay was configured in voice-port configuration mode. For Cisco IOS Release 12.1(5)T and later releases, in most cases playout delay should be configured in dial-peer configuration mode on the VoIP dial peer that is on the receiving end of the voice traffic that is to be buffered. This dial peer senses network conditions and relays them to the DSPs, which adjust the jitter buffer as necessary. When multiple applications are configured on the gateway, playout delay should be configured in dial-peer configuration mode. When there are numerous dial peers to configure, it might be simpler to configure playout delay on a voice port. If there are conflicting playout delay configurations on a voice port and also on a dial peer, the dial peer configuration takes precedence.

To configure the playout delay jitter buffer, use the following commands beginning in dial-peer or voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# playout-delay mode {adaptive | fixed}`	Determines the mode in which the jitter buffer will operate for calls on this voice port. The keywords are as follows: • **adaptive**—Adjusts the jitter buffer size and amount of playout delay during a call based on current network conditions. • **fixed**—Defines the jitter buffer size as fixed so that the playout delay does not adjust during a call. A constant playout delay is added. The default is **adaptive**.

	Command	Purpose
Step 2	`Router(config-voiceport)# playout-delay {nominal value | maximum value | minimum {default | low | high}}`	Tunes the playout buffer to accommodate packet jitter caused by switches in the WAN. The keywords and arguments are as follows: • **nominal**—Defines the amount of playout delay applied at the beginning of a call by the jitter buffer in the gateway. In fixed mode, this is also the maximum size of the jitter buffer throughout the call. • *value*—Specifies the range that depends on type of DSP and configured codec complexity. For medium codec complexity, the range is from 0 to 150 ms. For high codec complexity and DSPs that do not support codec complexity, the range is from 0 to 250 ms.
		• **maximum** (adaptive mode only)—Specifies the jitter buffer's upper limit (80ms), or the highest value to which the adaptive delay is set. • **minimum** (adaptive mode only)—Specifies the jitter buffer's lower limit (10 ms), or the lowest value to which the adaptive delay is set. • **default**—Specifies 40 ms.

Echo Adjustment

Echo is the sound of your own voice reverberating in the telephone receiver while you are talking. When timed properly, echo is not a problem in the conversation; however, if the echo interval exceeds approximately 25 milliseconds, it is distracting. Echo is controlled by echo cancellers.

In the traditional telephony network, echo is generally caused by an impedance mismatch when the four-wire network is converted to the two-wire local loop. In voice packet-based networks, echo cancellers are built into the low-bit rate codecs and are operated on each DSP.

By design, echo cancellers are limited by the total amount of time they wait for the reflected speech to be received, which is known as an echo trail. The echo trail is normally 32 milliseconds. In Cisco System's voice implementations, echo cancellers are enabled using the **echo-cancel enable** command, and echo trails are configured using the **echo-cancel coverage** command.

To configure parameters related to the echo canceller, use the following commands beginning in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# echo-cancel enable`	Enables the cancellation of voice that is sent and received on the same interface. Echo cancellation coverage must also be configured. The default is that echo cancellation is enabled. **Note** Not valid for four-wire E&M interfaces. Use **no echo-cancel enable** to disable the feature.
Step 2	`Router(config-voiceport)# echo-cancel coverage {8 | 16 | 24 | 32}`	Adjusts the echo canceller by the specified number of milliseconds. The default is 16.
Step 3	`Router(config-voiceport)# non-linear`	Enables nonlinear processing (residual echo suppression) in the echo canceler, which shuts off any signal if no near-end speech is detected. Echo cancelling must be enabled for this feature. The default is that nonlinear processing is enabled.

Voice Level Adjustment

As much as possible, it is desirable to achieve a uniform input decibel level to the packet voice network in order to limit or eliminate any voice distortion due to incorrect input and output decibel levels. Adjustments to levels may be required by the type of equipment connected to the network or by local country-specific conditions.

Incorrect input or output levels can cause echo, as can an impedance mismatch. Too much input gain can cause clipped or fuzzy voice quality. If the output level is too high at the remote router's voice port, the local caller will hear echo. If the local router's voice port input decibel level is too high, the remote side will hear clipping. If the local router's voice port input decibel level is too low, or the remote router's output level is too low, the remote side voice can be distorted at a very low volume and DTMF may be missed.

Use the **input gain** and **output attenuation** commands to adjust voice levels, and the **impedance** command to set the impedance value to match that of the voice circuit to which the voice port connects.

To change parameters related to voice levels, use the following commands as appropriate, in voice-port configuration mode:

	Command	Purpose
Step 1	`Router(config-voiceport)# input gain` *value*	Specifies, in decibels, the amount of gain to be inserted at the receiver side of the interface, increasing or decreasing the signal. After an input gain setting is changed, the voice call must be disconnected and reestablished before the changes take effect. The *value* argument is any integer from –6 to 14. The default is 0.
Step 2	`Router(config-voiceport)# output attenuation` *value*	Specifies the amount of attenuation in decibels at the transmit side of the interface, decreasing the signal. A system-wide loss plan can be implemented using the **input gain** and **output attenuation** commands. The default value for this command assumes that a standard transmission loss plan is in effect, meaning that normally there must be –6 dB attenuation between phones. The *value* argument is any integer from –6 to 14. The default is 0.
Step 3	`Router(config-voiceport)# impedance {600c | 600r | 900c | complex1 | complex2}`	Specifies the terminating impedance of a voice port interface, which needs to match the specifications from the specific telephony system to which it is connected. • **600c**—Specifies 600 ohms complex. • **600r**—Specifies 600 ohms real. • **900c**—Specifies 900 ohms complex. • **complex1**—Specifies Complex 1. • **complex2**—Specifies Complex 2. The default is 600r.

	Command	Purpose
Step 4	`Router(config-voiceport)# loss-plan {plan1 | plan2 | plan5 | plan6 | plan7 | plan8 | plan9}`	(Cisco MC3810 multiservice concentrators FXO or FXS analog voice ports only) Specifies the analog-to-digital gain offset loss plan. For definitions of each plan, see the *Cisco IOS Voice, Video, and Fax Command Reference*. The default is the **plan1** keyword.
Step 5	`Router(config-voiceport)# idle-voltage {high | low}`	(Cisco MC3810 multiservice concentrators analog FXS ports only) Specifies the talk-battery (tip-to-ring) voltage condition when the port is idle. The keywords are as follows: • **high**—Specifies that the voltage is high (–48V). • **low**—Specifies that the voltage is low (–24V) and is the default.

Verifying Analog and Digital Voice-Port Configurations

After configuring the voice ports on your router, perform the following steps to verify proper operation:

Step 1 Pick up the handset of an attached telephony device and check for a dial tone.

Step 2 If you have dial tone, check for DTMF detection. If the dial tone stops when you dial a digit, then the voice port is most likely configured properly.

Step 3 To identify port numbers of voice interfaces installed in your router, use the **show voice port summary** command. For examples of the output, see the "show voice port summary Command Examples" section on page 134.

Step 4 To verify voice-port parameter settings, use the **show voice port** command with the appropriate syntax from Table 12. For sample output, see the "show voice port Command Examples" section on page 135.

Table 12 Show Voice Port Command Syntax

Platform	Voice Port Type	Command Syntax
Cisco 1750	Analog	**show voice port** [*slot/port* | **summary**]
Cisco 2600 series Cisco 3600 series	Analog	**show voice port** [*slot/port* | **summary**]
	Digital	**show voice port** [*slot/port:ds0-group-no* | **summary**]
Cisco MC3810	Analog	**show voice port** [*slot/port* | **summary**]
	Digital	**show voice port** [*slot:ds0-group-no* | **summary**]
Cisco AS5300	Digital	**show voice-port** *controller:ds0-group-no*
Cisco AS5800	Digital	**show voice-port** {*shelf/slot/port:ds0-group-no*}
Cisco 7200 series	Digital	**show voice port** {*slot/port-adapter:ds0-group-no*}
Cisco 7500 series	Digital	**show voice port** {*slot/port-adapter/slot:ds0-group-no*}

Step 5 For digital T1/E1 connections, to verify the codec complexity configuration, use the **show running-config** command to display the current voice-card setting. If medium complexity is specified, the codec complexity setting is not displayed. If high complexity is specified, the setting codec complexity high is displayed. The following example shows an excerpt from the command output when high complexity has been specified:

```
Router# show running-config
.
.
.
hostname router-alpha

voice-card 0
 codec complexity high
.
.
.
```

Step 6 For digital T1/E1 connections, to verify that the controller is up and that no alarms have been reported, and to display information about clock sources and other controller settings, use the **show controller** command. For output examples, see the "show controller Command Examples" section on page 139.

```
Router# show controller {t1 | e1} controller-number
```

Step 7 To display voice-channel configuration information for all DSP channels, use the **show voice dsp** command. For output examples, see the "show voice dsp Command Examples" section on page 140.

```
Router# show voice dsp
```

Step 8 To verify the call status for all voice ports, use the **show voice call summary** command. For output examples, see the "show voice call summary Command Examples" section on page 141.

```
Router# show voice call summary
```

Step 9 To display the contents of the active call table, which shows all of the calls currently connected through the router or concentrator, use the **show call active voice** command. For output examples, see the "show call active voice Command Example" section on page 141.

```
Router# show call active voice
```

Step 10 To display the contents of the call history table, use the **show call history voice** command. To limit the display to the last calls connected through this router, use the keyword **last** and define the number of calls to be displayed with the argument *number*. To limit the display to a shortened version of the call history table, use the **brief** keyword. For output examples, see the "show call history voice Command Example" section on page 142.

```
Router# show call history voice [last | number | brief]
```

show voice port summary Command Examples

In the following sections, output examples of the following types are shown:

- Cisco 3640 Router Analog Voice Port
- Cisco MC3810 Multiservice Concentrator Digital Voice Port

Cisco 3640 Router Analog Voice Port

The following output is from a Cisco 3640 router:

```
Router# show voice port summary
                                        IN      OUT
PORT    CH SIG-TYPE    ADMIN OPER STATUS   STATUS   EC
====== == ========== ===== ==== ======== ======== ==
2/0/0   -- e&m-wnk    up    dorm idle     idle     y
2/0/1   -- e&m-wnk    up    dorm idle     idle     y
2/1/0   -- fxs-ls     up    dorm on-hook  idle     y
2/1/1   -- fxs-ls     up    dorm on-hook  idle     y
```

Cisco MC3810 Multiservice Concentrator Digital Voice Port

The following output is from a Cisco MC3810 multiservice concentrator:

```
Router# show voice port summary
                                        IN      OUT
PORT    CH SIG-TYPE    ADMIN OPER STATUS   STATUS   EC
====== == ========== ===== ==== ======== ======== ==
0:17    18 fxo-ls     down  down idle     on-hook  y
0:18    19 fxo-ls     up    dorm idle     on-hook  y
0:19    20 fxo-ls     up    dorm idle     on-hook  y
0:20    21 fxo-ls     up    dorm idle     on-hook  y
0:21    22 fxo-ls     up    dorm idle     on-hook  y
0:22    23 fxo-ls     up    dorm idle     on-hook  y
0:23    24 e&m-imd    up    dorm idle     idle     y
1/1     -- fxs-ls     up    dorm on-hook  idle     y
1/2     -- fxs-ls     up    dorm on-hook  idle     y
1/3     -- e&m-imd    up    dorm idle     idle     y
1/4     -- e&m-imd    up    dorm idle     idle     y
1/5     -- fxo-ls     up    dorm idle     on-hook  y
1/6     -- fxo-ls     up    dorm idle     on-hook  y
```

show voice port Command Examples

In the following sections, output examples of the following types are shown:

Cisco 3600 Series Router Analog E&M Voice Port

The following output is from a Cisco 3600 series router analog E&M voice port:

```
Router# show voice port 1/0/0

E&M Slot is 1, Sub-unit is 0, Port is 0
 Type of VoicePort is E&M
 Operation State is unknown
 Administrative State is unknown
 The Interface Down Failure Cause is 0
 Alias is NULL
```

```
 Noise Regeneration is disabled
 Non Linear Processing is disabled
 Music On Hold Threshold is Set to 0 dBm
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is disabled
 Echo Cancel Coverage is set to 16ms
 Connection Mode is Normal
 Connection Number is
 Initial Time Out is set to 0 s
 Interdigit Time Out is set to 0 s
 Analog Info Follows:
 Region Tone is set for northamerica
 Currently processing none
 Maintenance Mode Set to None (not in mtc mode)
 Number of signaling protocol errors are 0

 Voice card specific Info Follows:
 Signal Type is wink-start
 Operation Type is 2-wire
 Impedance is set to 600r Ohm
 E&M Type is unknown
 Dial Type is dtmf
 In Seizure is inactive
 Out Seizure is inactive
 Digit Duration Timing is set to 0 ms
 InterDigit Duration Timing is set to 0 ms
 Pulse Rate Timing is set to 0 pulses/second
 InterDigit Pulse Duration Timing is set to 0 ms
 Clear Wait Duration Timing is set to 0 ms
 Wink Wait Duration Timing is set to 0 ms
 Wink Duration Timing is set to 0 ms
 Delay Start Timing is set to 0 ms
 Delay Duration Timing is set to 0 ms
```

Cisco 3600 Series Router Analog FXS Voice Port

The following output is from a Cisco 3600 series router analog FXS voice port:

```
Router# show voice port 1/2

Voice port 1/2 Slot is 1, Port is 2
 Type of VoicePort is FXS
 Operation State is UP
 Administrative State is UP
 No Interface Down Failure
 Description is not set
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is enabled
 Echo Cancel Coverage is set to 8 ms
 Connection Mode is normal
 Connection Number is not set
 Initial Time Out is set to 10 s
 Interdigit Time Out is set to 10 s
 Coder Type is g729ar8
 Companding Type is u-law
 Voice Activity Detection is disabled
 Ringing Time Out is 180 s
 Wait Release Time Out is 30 s
 Nominal Playout Delay is 80 milliseconds
```

```
 Maximum Playout Delay is 160 milliseconds

 Analog Info Follows:
 Region Tone is set for northamerica
 Currently processing Voice
 Maintenance Mode Set to None (not in mtc mode)
 Number of signaling protocol errors are 0
 Impedance is set to 600r Ohm
 Analog interface A-D gain offset = -3 dB
 Analog interface D-A gain offset = -3 dB
 Voice card specific Info Follows:
 Signal Type is loopStart
 Ring Frequency is 20 Hz
 Hook Status is On Hook
 Ring Active Status is inactive
 Ring Ground Status is inactive
 Tip Ground Status is active
 Digit Duration Timing is set to 100 ms
 InterDigit Duration Timing is set to 100 ms
 Ring Cadence are [20 40] * 100 msec
 InterDigit Pulse Duration Timing is set to 500 ms
```

Cisco 3600 Series Router Digital E&M Voice Port

The following output is from a Cisco 3600 series router digital E&M voice port:

```
Router# show voice port 1/0:1

receEive and transMit Slot is 1, Sub-unit is 0, Port is 1
 Type of VoicePort is E&M
 Operation State is DORMANT
 Administrative State is UP
 No Interface Down Failure
 Description is not set
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 Music On Hold Threshold is Set to -38 dBm
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is enabled
 Echo Cancel Coverage is set to 8 ms
 Connection Mode is normal
 Connection Number is not set
 Initial Time Out is set to 10 s
 Interdigit Time Out is set to 10 s
 Region Tone is set for US
```

Cisco AS5300 Universal Access Server T1 CAS Voice Port

The following output is from a Cisco AS5300 universal access server T1 CAS voice port:

```
Router# show voice port

DS0 Group 1:0 - 1:0
 Type of VoicePort is CAS
 Operation State is DORMANT
 Administrative State is UP
 No Interface Down Failure
 Description is not set
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 Music On Hold Threshold is Set to -38 dBm
```

```
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is enabled
 Echo Cancel Coverage is set to 8 ms
 Playout-delay Mode is set to default
 Playout-delay Nominal is set to 60 ms
 Playout-delay Maximum is set to 200 ms
 Connection Mode is normal
 Connection Number is not set
 Initial Time Out is set to 10 s
 Interdigit Time Out is set to 10 s
 Call-Disconnect Time Out is set to 60 s
 Ringing Time Out is set to 180 s
 Companding Type is u-law
 Region Tone is set for US
 Wait Release Time Out is 30 s
 Station name None, Station number None

 Voice card specific Info Follows:

 DS0 channel specific status info:
                                IN      OUT
    PORT   CH SIG-TYPE   OPER STATUS   STATUS    TIP     RING
```

Cisco 7200 Series Router Digital E&M Voice Port

The following output is from a Cisco 7200 series router digital E&M voice port:

```
Router# show voice port 1/0:1

receEive and transMit Slot is 1, Sub-unit is 0, Port is 1  << voice-port 1/0:1

 Type of VoicePort is E&M

 Operation State is DORMANT

 Administrative State is UP

 No Interface Down Failure
 Description is not set
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 Music On Hold Threshold is Set to -38 dBm
 In Gain is Set to 0 dB

 Out Attenuation is Set to 0 dB

 Echo Cancellation is enabled

 Echo Cancel Coverage is set to 8 ms
 Connection Mode is normal
 Connection Number is not set
 Initial Time Out is set to 10 s

 Interdigit Time Out is set to 10 s

 Region Tone is set for US
```

show controller Command Examples

In the following sections, output examples of the following types are shown:

Cisco 3600 Series Router T1 Controller

The following output is from a Cisco 3600 series router with a T1 controller:

```
Router# show controller T1 1/1/0

T1 1/0/0 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  alarm-trigger is not set
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (180 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Cisco MC3810 Multiservice Concentrator E1 Controller

The following output is from a Cisco MC3810 multiservice concentrator with an E1 controller:

```
Router# show controller E1 1/0

E1 1/0 is up.
  Applique type is Channelized E1
  Cablelength is short 133
  Description: E1 WIC card Alpha
  No alarms detected.
  Framing is CRC4, Line Code is HDB3, Clock Source is Line Primary.
  Data in current interval (1 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Cisco AS5800 Universal Access Server T1 Controller

The following output is from a Cisco AS5800 universal access server with a T1 controller:

```
Router# show controller t1 2

 T1 2 is up.
   No alarms detected.
   Version info of slot 0:  HW: 2, Firmware: 16, PLD Rev: 0

 Manufacture Cookie Info:
  EEPROM Type 0x0001, EEPROM Version 0x01, Board ID 0x42,
  Board Hardware Version 1.0, Item Number 73-2217-4,
  Board Revision A0, Serial Number 06467665,
  PLD/ISP Version 0.0, Manufacture Date 14-Nov-1997.

    Framing is ESF, Line Code is B8ZS, Clock Source is Internal.
```

```
  Data in current interval (269 seconds elapsed):
   0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

show voice dsp Command Examples

The following output is from a Cisco 3640 router when a digital voice port is configured:

```
Router# show voice dsp

TYPE DSP CH CODEC    VERS STATE STATE   RST AI PORT    TS ABORT   TX/RX-PAK-CNT
==== === == ======== ==== ===== ======= === == ======= == ===== ===============
C549 010 00 g729r8    3.3 busy  idle      0  0 1/015   1     0     67400/85384
         01 g729r8     .8 busy  idle      0  0 1/015   7     0     67566/83623
         02 g729r8        busy  idle      0  0 1/015  13     0     65675/81851
         03 g729r8        busy  idle      0  0 1/015  20     0     65530/83610
C549 011 00 g729r8    3.3 busy  idle      0  0 1/015   2     0     66820/84799
         01 g729r8     .8 busy  idle      0  0 1/015   8     0     59028/66946
         02 g729r8        busy  idle      0  0 1/015  14     0     65591/81084
         03 g729r8        busy  idle      0  0 1/015  21     0     66336/82739
C549 012 00 g729r8    3.3 busy  idle      0  0 1/015   3     0     59036/65245
         01 g729r8     .8 busy  idle      0  0 1/015   9     0     65826/81950
         02 g729r8        busy  idle      0  0 1/015  15     0     65606/80733
         03 g729r8        busy  idle      0  0 1/015  22     0     65577/83532
C549 013 00 g729r8    3.3 busy  idle      0  0 1/015   4     0     67655/82974
         01 g729r8     .8 busy  idle      0  0 1/015  10     0     65647/82088
         02 g729r8        busy  idle      0  0 1/015  17     0     66366/80894
         03 g729r8        busy  idle      0  0 1/015  23     0     66339/82628
C549 014 00 g729r8    3.3 busy  idle      0  0 1/015   5     0     68439/84677
         01 g729r8     .8 busy  idle      0  0 1/015  11     0     65664/81737
         02 g729r8        busy  idle      0  0 1/015  18     0     65607/81820
         03 g729r8        busy  idle      0  0 1/015  24     0     65589/83889
C549 015 00 g729r8    3.3 busy  idle      0  0 1/015   6     0     66889/83331
         01 g729r8     .8 busy  idle      0  0 1/015  12     0     65690/81700
         02 g729r8        busy  idle      0  0 1/015  19     0     66422/82099
         03 g729r8        busy  idle      0  0 1/015  25     0     65566/83852

Router# show voice dsp

TYPE DSP CH CODEC    VERS STATE STATE   RST AI PORT    TS ABORT   TX/RX-PAK-CNT
==== === == ======== ==== ===== ======= === == ======= == ===== ===============
C549 007 00 {medium}  3.3 IDLE  idle      0  0 1/0:1    4     0             0/0
                      .13
C549 008 00 {medium}  3.3 IDLE  idle      0  0 1/0:1    5     0             0/0
                      .13
C549 009 00 {medium}  3.3 IDLE  idle      0  0 1/0:1    6     0             0/0
                      .13
C549 010 00 {medium}  3.3 IDLE  idle      0  0 1/0:1    7     0             0/0
                      .13
C549 011 00 {medium}  3.3 IDLE  idle      0  0 1/0:1    8     0             0/0
                      .13
C549 012 00 {medium}  3.3 IDLE  idle      0  0 1/0:1    9     0             0/0
                      .13
C542 001 01 g711ulaw  3.3 IDLE  idle      0  0 2/0/0          0         512/519
                      .13
C542 002 01 g711ulaw  3.3 IDLE  idle      0  0 2/0/1          0         505/502
                      .13
C542 003 01 g711alaw  3.3 IDLE  idle      0  0 2/1/0          0     28756/28966
                      .13
C542 004 01 g711ulaw  3.3 IDLE  idle      0  0 2/1/1          0         834/838
                      .13
```

show voice call summary Command Examples

In the following sections, output examples of the following types are shown:

- Cisco MC3810 Multiservice Concentrator Analog Voice Port
- Cisco 3600 Series Router Digital Voice Port

Cisco MC3810 Multiservice Concentrator Analog Voice Port

The following output is from a Cisco MC3810 multiservice concentrator:

```
Router# show voice call summary

PORT      CODEC    VAD VTSP STATE            VPM STATE
========= ======== === ==================== ========================
1/1       g729r8    y  S_CONNECT            FXSLS_CONNECT
1/2       -         -  -                    FXSLS_ONHOOK
1/3       -         -  -                    EM_ONHOOK
1/4       -         -  -                    EM_ONHOOK
1/5       -         -  -                    FXOLS_ONHOOK
1/6       -         -  -                    FXOLS_ONHOOK
```

Cisco 3600 Series Router Digital Voice Port

The following output is from a Cisco 3600 series router:

```
Router# show voice call summary

PORT      CODEC    VAD VTSP STATE            VPM STATE
========= ======== === ==================== ========================
1/015.1  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.2  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.3  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.4  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.5  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.6  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.7  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.8  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.9  g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.10 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.11 g729r8    y  S_CONNECT            S_TSP_CONNECT
1/015.12 g729r8    y  S_CONNECT            S_TSP_CONNECT
```

show call active voice Command Example

The following output is from a Cisco 7200 series router:

```
Router# show call active voice

GENERIC:
SetupTime=94523746 ms
Index=448
PeerAddress=##73072

PeerSubAddress=
PeerId=70000

PeerIfIndex=37
```

```
LogicalIfIndex=0
ConnectTime=94524043
DisconectTime=94546241
CallOrigin=1

ChargedUnits=0
InfoType=2
TransmitPackets=6251
TransmitBytes=125020
ReceivePackets=3300
ReceiveBytes=66000
VOIP:
ConnectionId[0x142E62FB 0x5C6705AF 0x0 0x385722B0]
RemoteIPAddress=171.68.235.18

RemoteUDPPort=16580

RoundTripDelay=29 ms

SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
SessionProtocol=cisco
SessionTarget=ipv4:171.68.235.18
OnTimeRvPlayout=63690
GapFillWithSilence=0 ms

GapFillWithPrediction=180 ms

GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=70 ms
LoWaterPlayoutDelay=30 ms
ReceiveDelay=40 ms
LostPackets=0 ms

EarlyPackets=1 ms

LatePackets=18 ms

VAD = disabled

CoderTypeRate=g729r8

CodecBytes=20

cvVoIPCallHistoryIcpif=0
SignalingType=cas
```

show call history voice Command Example

The following output is from a Cisco 7200 series router:

```
Router# show call history voice

GENERIC:
SetupTime=94893250 ms
Index=450
PeerAddress=##52258

PeerSubAddress=
PeerId=50000
```

```
PeerIfIndex=35
LogicalIfIndex=0
DisconnectCause=10

DisconnectText=normal call clearing.

ConnectTime=94893780
DisconectTime=95015500
CallOrigin=1

ChargedUnits=0
InfoType=2
TransmitPackets=32258
TransmitBytes=645160
ReceivePackets=20061
ReceiveBytes=401220
VOIP:
ConnectionId[0x142E62FB 0x5C6705B3 0x0 0x388F851C]
RemoteIPAddress=171.68.235.18

RemoteUDPPort=16552

RoundTripDelay=23 ms

SelectedQoS=best-effort
tx_DtmfRelay=inband-voice
SessionProtocol=cisco
SessionTarget=ipv4:171.68.235.18
OnTimeRvPlayout=398000
GapFillWithSilence=0 ms

GapFillWithPrediction=1440 ms

GapFillWithInterpolation=0 ms
GapFillWithRedundancy=0 ms
HiWaterPlayoutDelay=97 ms
LoWaterPlayoutDelay=30 ms
ReceiveDelay=49 ms
LostPackets=1 ms
EarlyPackets=1 ms

LatePackets=132 ms

VAD = disabled

CoderTypeRate=g729r8

CodecBytes=20
cvVoIPCallHistoryIcpif=0
```

Troubleshooting Analog and Digital Voice Port Configurations

The following sections will assist in analyzing and troubleshooting voice port problems:

Troubleshooting Chart

Table 13 lists some problems you might encounter after configuring voice ports and has some suggested remedies.

Table 13 Troubleshooting Voice Port Configurations

Problem	Suggested Action
No connectivity	Ping the associated IP address to confirm connectivity. If you cannot successfully ping your destination, refer to the *Cisco IOS IP Configuration Guide*.
No connectivity	Enter the **show controller t1** or **show controller e1** command with the controller number for the voice port you are troubleshooting. This will tell you: • If the controller is up. If it is not, use the **no shutdown** command to make it active. • Whether alarms have been reported. • What parameter values have been set for the controller (framing, clock source, line code, cable length). If these values do not match those of the telephony connection you are making, reconfigure the controller. See the "show controller Command Examples" section on page 139 for output.
No connectivity	Enter the **show voice port** command with the voice port number that you are troubleshooting, which will tell you: • If the voice port is up. If it is not, use the **no shutdown** command to make it active. • What parameter values have been set for the voice port, including default values (these do not appear in the output for the **show running-config** command). If these values do not match those of the telephony connection you are making, reconfigure the voice port. See the "show voice port Command Examples" section on page 135 for sample output.
Telephony device buzzes or does not ring	Use the **show voice port** command to confirm that ring frequency is configured correctly. It must match the connected telephony equipment and may be country-dependent.

Table 13 Troubleshooting Voice Port Configurations (continued)

Problem	Suggested Action
Distorted speech	Use the **show voice port** command to confirm the **cptone** keyword setting (also called *region tone*) is US. Setting a wrong cptone could result in faulty voice reproduction during analog-to-digital or digital-to-analog conversions.
Music on hold is not heard	Reduce the music-threshold level.
Background noise is not heard	Enable the **comfort-noise** command.
Long pauses occur in conversation; like speaking on a walkie-talkie	Overall delay is probably excessive; the standard for adequate voice quality is 150 ms one-way transit delay. Measure delay by using ping tests at various times of the day with different network traffic loads. If delay must be reduced, areas to examine include propagation delay of signals between the sending and receiving endpoints, voice encoding delay, and the voice packetization time for various VoIP codecs.
Jerky or choppy speech	Variable delay, or jitter, is being introduced by congestion in the packet network. Two possible remedies are to: • Reduce the amount of congestion in your packet network. Pings between VoIP endpoints will give an idea of the round-trip delay of a link, which should never exceed 300 ms. Network queuing and dropped packets should also be examined. • Increase the size of the jitter buffer with the **playout-delay** command. (See the "Jitter Adjustment" section on page 128.)
Clipped or fuzzy speech	Reduce input gain. (See the "Voice Level Adjustment" section on page 132.)
Clipped speech	Reduce the input level at the listener's router. (See the "Voice Level Adjustment" section on page 132.)
Volume too low or missed DTMF	Increase speaker's output level or listener's input level. (See the "Voice Level Adjustment" section on page 132.)
Echo interval is greater than 25 ms (sounds like a separate voice)	Configure the **echo-cancel enable** command and increase the value for the **echo-cancel coverage** keyword. (See the "Echo Adjustment" section on page 130.)
Too much echo	Reduce the output level at the speaker's voice port. (See the "Voice Level Adjustment" section on page 132.)

Voice Port Testing Commands

These commands allow you to force voice ports into specific states for testing. They require the use of Cisco IOS Release 12.0(7)XK or 12.1(2)T or a later release, and they apply only to Cisco 2600 and 3600 series routers, and to Cisco MC3810 multiservice concentrators. The following types of voice-port tests are covered:

- Detector-Related Function Tests, page 146
- Loopback Function Tests, page 148
- Tone Injection Tests, page 149
- Relay-Related Function Tests, page 150
- Fax/Voice Mode Tests, page 150

Detector-Related Function Tests

Using the **test voice port detector** command, you are able to force a particular detector into an on or off state, perform tests on the detector, and then return the detector to its original state.

To configure this feature, use the following commands in privileged EXEC mode:

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} {on | off}`
	
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} {on | off}`
	
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} {on | off}`
		Identifies the voice port you want to test. Enter a keyword for the detector under test and specify whether to force it to the on or off state.
		
		Note For each signaling type (E&M, FXO, FXS), only the applicable keywords are displayed. The **disable** keyword is displayed only when a detector is in the forced state.

	Command	Purpose
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports `Router# test voice port slot:ds0-group detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} {on | off}`	

Step 2	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} disable`
	
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} disable`
	
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} disable`
	
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group detector {m-lead | battery-reversal | loop-current | ring | tip-ground | ring-ground | ring-trip} disable`
		Identifies the voice port on which you want to end the test. Enter a keyword for the detector under test and the keyword **disable** to end the forced state.
		
		Note For each signaling type (E&M, FXO, FXS), only the applicable keywords are displayed. The **disable** keyword is displayed only when a detector is in the forced state.

Loopback Function Tests

To establish loopbacks on a voice port, use the following commands in privileged EXEC mode:

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port loopback {local | network}`
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group loopback {local | network}`
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector loopback {local | network}`
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group loopback {local | network}`
		Identifies the voice port you want to test and enters a keyword for the loopback direction.
		Note A call must be established on the voice port under test.

Step 2	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port loopback disable`
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group loopback disable`
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector loopback disable`
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group loopback disable`
		Identifies the voice port on which you want to end the test and enters the keyword disable to end the loopback.

Tone Injection Tests

To inject a test tone into a voice port, use the following commands in privileged EXEC mode:

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port inject-tone {local | network} {1000hz | 2000hz | 200hz | 3000hz | 300hz | 3200hz | 3400hz | 500hz | quiet}`
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group inject-tone {local | network} {1000hz | 2000hz | 200hz | 3000hz | 300hz | 3200hz | 3400hz | 500hz | quiet}`
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector inject-tone {local | network} {1000hz | 2000hz | 200hz | 3000hz | 300hz | 3200hz | 3400hz | 500hz | quiet}`
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group inject-tone {local | network} {1000hz | 2000hz | 200hz | 3000hz | 300hz | 3200hz | 3400hz | 500hz | quiet}`
		Identifies the voice port you want to test and enter keywords for the direction to send the test tone and for the frequency of the test tone.
		Note A call must be established on the voice port under test.

Step 2	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port inject-tone disable`
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group inject-tone disable`
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector inject-tone disable`
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group inject-tone disable`
		Identifies the voice port on which you want to end the test and enter the keyword **disable** to end the test tone.
		Note The **disable** keyword is available only if a test condition is already activated.

Relay-Related Function Tests

To test relay-related functions on a voice port, use the following commands in privileged EXEC mode:

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} {on | off}`
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} {on | off}`
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} {on | off}`
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} {on | off}`
		Identifies the voice port you want to test. Enter a keyword for the relay under test and specify whether to force it to the on or off state.
		Note For each signaling type (E&M, FXO, FXS), only the applicable keywords are displayed. The **disable** keyword is displayed only when a relay is in the forced state.

Step 2	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} disable`
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} disable`
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} disable`
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group relay {e-lead | loop | ring-ground | battery-reversal | power-denial | ring | tip-ground} disable`
		Identifies the voice port on which you want to end the test. Enter a keyword for the relay under test, and the keyword **disable** to end the forced state.
		Note For each signaling type (E&M, FXO, FXS), only the applicable keywords are displayed. The **disable** keyword is displayed only when a relay is in the forced state.

Fax/Voice Mode Tests

The **test voice port switch fax** command forces a voice port into fax mode for testing. After you enter this command, you can use the **show voice call** or **show voice call summary** command to check whether the voice port is able to operate in fax mode. If no fax data is detected by the voice port, the voice port remains in fax mode for 30 seconds and then reverts automatically to voice mode.

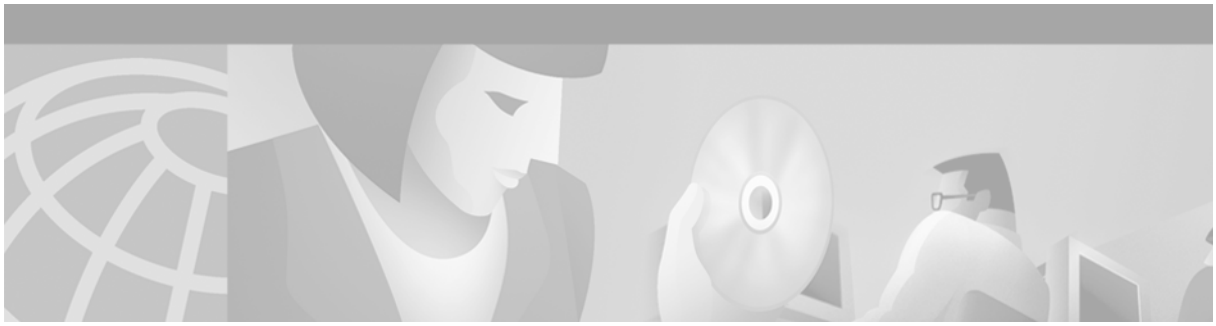
The **disable** keyword ends the forced mode switch; however, the fax mode ends automatically after 30 seconds. The disable keyword is available only while the voice port is in fax mode.

To force a voice port into fax mode and return it to voice mode, use the following commands in privileged EXEC mode:

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port switch fax`
	
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group switch fax`
	
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector switch fax`
	
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group switch fax`
		Identifies the voice port you want to test. Enter the keyword **fax** to force the voice port into fax mode.

Step 2	**Cisco 2600 and 3600 Series Routers Analog Voice Ports**
	`Router# test voice port slot/subunit/port switch disable`
	
	Cisco 2600 and 3600 Series Routers Digital Voice Ports
	`Router# test voice port slot/port:ds0-group switch disable`
	
	Cisco MC3810 Multiservice Concentrators Analog Voice Ports
	`Router# test voice port slot/port detector switch disable`
	
	Cisco MC3810 Multiservice Concentrators Digital Voice Ports
	`Router# test voice port slot:ds0-group switch disable`
		Identifies the voice port on which you want to end the test. Enter the keyword **disable** to return the voice port to voice mode.



Configuring Dial Plans, Dial Peers, and Digit Manipulation

This chapter describes how to implement dial plans by configuring dial peers and using dial peer matching and digit manipulation features. This chapter contains the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Dial Plan Overview

A dial plan essentially describes the number and pattern of digits that a user dials to reach a particular telephone number. Access codes, area codes, specialized codes, and combinations of the number of digits dialed are all part of a dial plan. For instance, the North American Public Switched Telephone Network (PSTN) uses a 10-digit dial plan that includes a 3-digit area code and a 7-digit telephone number. Most PBXs support variable length dial plans that use 3 to 11 digits. Dial plans must comply with the telephone networks to which they connect. Only totally private voice networks that are not linked to the PSTN or to other PBXs can use any dial plan they choose.

Dial plans on Cisco routers are manually defined using dial peers. Dial peers are similar to static routes; they define where calls originate and terminate and what path the calls take through the network. Attributes within the dial peer determine which dialed digits the router collects and forwards to telephony devices.

Note If you are using Media Gateway Control Protocol (MGCP) or Simple Gateway Control Protocol (SGCP) on your call agent, you do not need to configure static dial peers. See the chapter "Configuring MGCP and Related Protocols" for more information.

The following sections provide an overview of basic dial peer concepts:

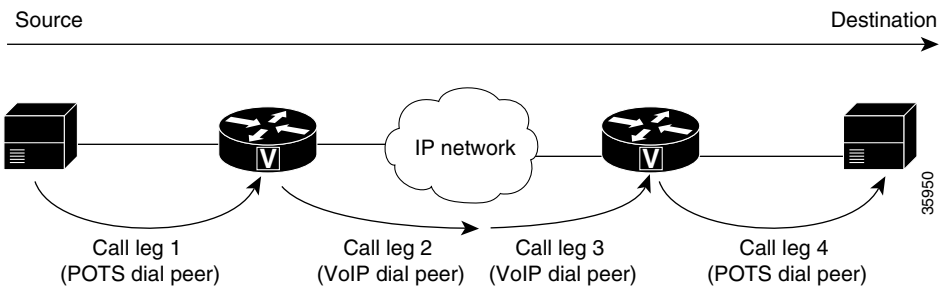
Note The illustrations and sample configurations in this section use VoIP; the same concepts also apply to Voice over Frame Relay (VoFR) and Voice over ATM (VoATM) networks.

Dial Peer Overview

Configuring dial peers is the key to setting up dial plans and implementing voice over a packet network. Dial peers are used to identify call source and destination endpoints and to define the characteristics applied to each call leg in the call connection.

A traditional voice call over the PSTN uses a dedicated 64K circuit end to end. In contrast, a voice call over the packet network is made up of discrete segments or call legs. A call leg is a logical connection between two routers or between a router and a telephony device. A voice call comprises four call legs, two from the perspective of the originating router and two from the perspective of the terminating router, as shown in Figure 39.

Figure 39 Dial Peer Call Legs



A dial peer is associated with each call leg. Attributes that are defined in a dial peer and applied to the call leg include codec, Quality of Service (QoS), voice activity detection (VAD), and fax rate. To complete a voice call, you must configure a dial peer for each of the four call legs in the call connection.

Depending on the call leg, a call is routed using one of the two types of dial peers:

- POTS—Dial peer that defines the characteristics of a traditional telephony network connection. POTS dial peers map a dialed string to a specific voice port on the local router, normally the voice port connecting the router to the local PSTN, PBX, or telephone.
- Voice-network—Dial peer that defines the characteristics of a packet network connection. Voice-network dial peers map a dialed string to a remote network device, such as the destination router that is connected to the remote telephony device.

  The specific type of voice-network dial peer depends on the packet network technology:

  - VoIP (Voice over IP)—Points to the IP address of the destination router that terminates the call.
  - VoFR (Voice over Frame Relay)—Points to the data-link connection identifier (DLCI) of the interface from which the call exits the router.
  - VoATM (Voice over ATM)—Points to the ATM virtual circuit for the interface from which the call exits the router.
  - MMoIP (Multimedia Mail over IP)—Points to the e-mail address of the SMTP server. This type of dial peer is used only for fax traffic. For more information, see the chapter "Configuring Fax Applications."

Both POTS and voice-network dial peers are needed to establish voice connections over a packet network.

Inbound and Outbound Dial Peers

Dial peers are used for both inbound and outbound call legs. It is important to remember that these terms are defined from the perspective of the router. An inbound call leg originates when an incoming call comes *to* the router. An outbound call leg originates when an outgoing call is placed *from* the router. Figure 40 illustrates call legs from the perspective of the originating router; Figure 41 illustrates call legs from the perspective of the terminating router.

Note Figure 40 and Figure 41 apply to voice calls that are being sent across the packet network. If the originating and terminating POTS interfaces share the same router or if the call requires hairpinning, then two POTS call legs are sufficient. See Figure 46 on page 162 for more information.

Figure 40 Call Legs from the Perspective of the Originating Router

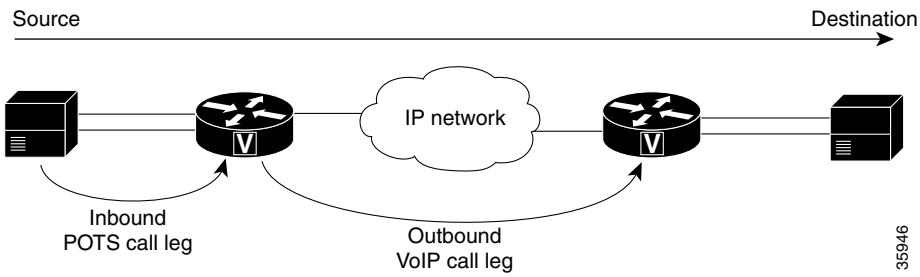
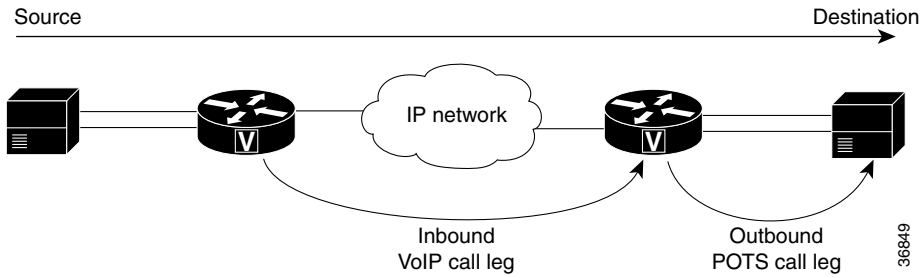


Figure 41 Call Legs from the Perspective of the Terminating Router



For inbound calls from a POTS interface that are destined for the packet network, the router matches a POTS dial peer for the inbound call leg and a voice-network dial peer, such as VoIP or VoFR, for the outbound leg. For inbound calls from the packet network, the router matches a POTS dial peer to terminate the call and a voice-network dial peer to apply features such as codec or QoS.

For inbound POTS call legs going to outbound voice-network dial peers, the router forwards all digits that it collects. On outbound POTS call legs, the router strips off explicitly matching digits and forwards any excess digits out the designated port. For specific information about how the router handles excess digits, see the "Two-Stage Dialing" section on page 173.

The following examples show basic configurations for POTS and VoIP dial peers:

```
dial-peer voice 1 pots
 destination-pattern 555....
 port 1/0:1

dial-peer voice 2 voip
 destination-pattern 555....
 session target ipv4:192.168.1.1
```

The router selects a dial peer for a call leg by matching the string that is defined by using the **answer-address**, **destination-pattern**, or **incoming called-number** command in the dial peer configuration. For specific information about how the router matches dial peers, see the "Dial Peer Overview" section on page 173.

Destination Pattern

The destination pattern associates a dialed string with a specific telephony device. It is configured in a dial peer by using the **destination-pattern** command. If the dialed string matches the destination pattern, the call is routed according to the voice port in POTS dial peers, or the session target in voice-network dial peers. For outbound voice-network dial peers, the destination pattern may also determine the dialed digits that the router collects and then forwards to the remote telephony interface, such as a PBX, a telephone, or the PSTN. You must configure a destination pattern for each POTS and voice-network dial peer that you define on the router.

The destination pattern can be either a complete telephone number or a partial telephone number with wildcard digits, represented by a period (.) character. Each "." represents a wildcard for an individual digit that the originating router expects to match. For example, if the destination pattern for a dial peer is defined as "555. . . .", then any dialed string beginning with 555, plus at least four additional digits, matches this dial peer.

In addition to the period (.), there are several other symbols that can be used as wildcard characters in the destination pattern. These symbols provide additional flexibility in implementing dial plans and decrease the need for multiple dial peers in configuring telephone number ranges.

Table 14 shows the wildcard characters that are supported in the destination pattern.

Table 14 Wildcard Symbols Used in Destination Patterns

Symbol	Description
.	Indicates a single-digit placeholder. For example, 555. . . . matches any dialed string beginning with 555, plus at least four additional digits.
[ ]	Indicates a range of digits. A consecutive range is indicated with a hyphen (-); for example, [5-7]. A nonconsecutive range is indicated with a comma (,); for example, [5,8]. Hyphens and commas can be used in combination; for example, [5-7,9]. **Note** Only single-digit ranges are supported. For example, [98-102] is invalid.
( )	Indicates a pattern; for example, 408(555). It is used in conjunction with the symbol ?, %, or +.
?	Indicates that the preceding digit occurred zero or one time. Enter **ctrl-v** before entering ? from your keyboard.
%	Indicates that the preceding digit occurred zero or more times. This functions the same as the "*" used in regular expression.
+	Indicates that the preceding digit occurred one or more times.
T	Indicates the interdigit timeout. The router pauses to collect additional dialed digits.

Note The period (.) is the only wildcard character that is supported for dial strings that are configured using the **answer-address** or **incoming called-number** commands.

Table 15 shows some examples of how these wildcard symbols are applied to the destination pattern and the dial string that results when dial string 4085551234 is matched to an outbound POTS dial peer. The wildcard symbols follow regular expression rules.

Table 15 Dial Peer Matching Examples Using Wildcard Symbols

	Destination Pattern	Dial String Translation	String After Stripping[1]
1	408555.+	408555, followed by one or more wildcard digits. This pattern implies that the string must contain at least 7 digits starting with 408555.	1234
2	408555.%	408555, followed by zero or more wildcard digits. This pattern implies that the string must contain at least 408555.	1234
3	408555+	40855, followed by 5 repeated one or more times.	1234
4	408555%	40855, followed by 5 repeated zero or more times. Any explicitly matching digit before the % symbol is not stripped off.	51234
5	408555?	40855, followed by 5 repeated zero or one time. Any explicitly matching digit before the ? symbol is not stripped off.	51234
6	40855[5-7].+	40855, followed by 5, 6, or 7, plus any digit repeated one or more times.	51234

Table 15 Dial Peer Matching Examples Using Wildcard Symbols (continued)

	Destination Pattern	Dial String Translation	String After Stripping[1]
7	40855[5-7].%	40855, followed by 5, 6, or 7, plus any digit repeated zero or more times.	51234
8	40855[5-7]+1234	40855, followed by 5, 6, or 7 repeated one or more times, followed by 1234.	51234
9	408(555)+1234	408, followed by 555, which may repeat one or more times, followed by 1234.	5551234

1. These examples apply only to one-stage dialing, where DID is enabled on the inbound POTS dial peer. If the router is using two-stage dialing and collecting digits one at a time as dialed, then the call is routed immediately after a dial peer is matched and any subsequent dialed digits are lost.

In addition to wildcard characters, the following characters can also be used in the destination pattern:

- Asterisk (*) and pound sign (#)—These characters on standard touch-tone dial pads can be used anywhere in the pattern. They can be used as the leading character (for example, *650), except on the Cisco 3600 series.
- Dollar sign ($)—Disables variable-length matching. Must be used at the end of the dial string.

The same destination pattern can be shared across multiple dial peers to form hunt groups. For information on building hunt groups, see the "Hunt Groups and Preferences" section on page 180.

For information on how the terminating router strips off digits after matching a destination pattern, see the "Digit Stripping on Outbound POTS Dial Peers" section on page 160.

Fixed- and Variable-Length Dial Plans

Fixed-length dialing plans, in which all the dial-peer destination patterns have a fixed length, are sufficient for most voice networks because the telephone number strings are of known lengths. Some voice networks, however, require variable-length dial plans, particularly for international calls, which use telephone numbers of different lengths.

If you enter the timeout T-indicator at the end of the destination pattern in an outbound voice-network dial peer, the router accepts a fixed-length dial string and then waits for additional dialed digits. The timeout character must be an uppercase T. The following dial-peer configuration shows how the T-indicator is set to allow variable-length dial strings:

```
dial-peer voice 1 voip
 destination-pattern 2222T
 session target ipv4:10.10.1.1
```

In the example above, the router accepts the digits 2222, and then waits for an unspecified number of additional digits. The router can collect up to 31 additional digits, as long as the interdigit timeout has not expired. When the interdigit timeout expires, the router places the call.

The default value for the interdigit timeout is 10 seconds. Unless the default value is changed, using the T-indicator adds 10 seconds to each call setup because the call is not attempted until the timer has expired (unless the # character is used as a terminator). You should therefore reduce the voice-port interdigit timeout value if you use variable-length dial plans. You can change the interdigit timeout by using the **timeouts inter-digit** voice-port command.

The calling party can immediately terminate the interdigit timeout by entering the # character. If the # character is entered while the router is waiting for additional digits, the # character is treated as a terminator; it is not treated as part of the dial string or sent across the network. But if the # character is entered before the router begins waiting for additional digits (meaning that the # is entered as part of the fixed-length destination pattern), then the # character is treated as a dialed digit.

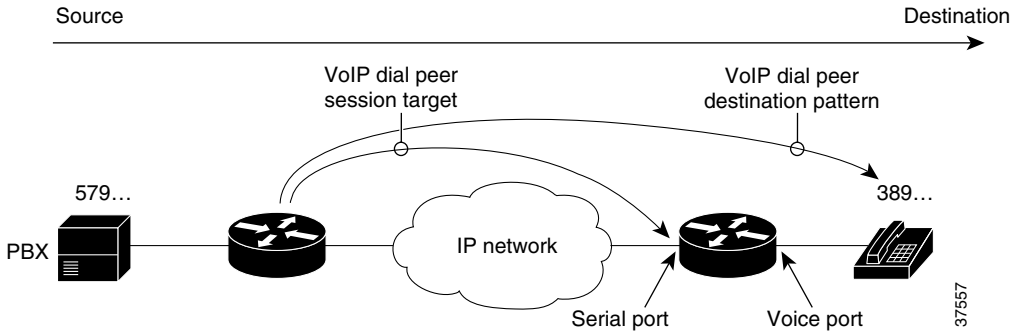
For example, if the destination pattern is configured as 2222. . . T, then the entire dialed string of 2222#9999 is collected, but if the dialed string is 2222#99#99, the #99 at the end of the dialed digits is not collected because the final # character is treated as a terminator. You can change the termination character by using the **dial-peer terminator** command.

Note In most cases, you must configure the T-indicator only when the router uses two-stage dialing. If Direct Inward Dialing (DID) is configured in the inbound POTS dial peer, the router uses one-stage dialing, which means that the full dialed string is used to match outbound dial peers. The only exception is when the **ISDN overlap-receiving** command is configured; the ISDN overlap-receiving feature requires the T-indicator.

Session Target

The session target is the network address of the remote router to which you want to send a call once a local voice-network dial peer is matched. It is configured in voice-network dial peers by using the **session target** command. For outbound dial peers, the destination pattern is the telephone number of the remote voice device that you want to reach. The session target represents the path to the remote router that is connected to that voice device. Figure 42 illustrates the relationship between the destination pattern and the session target, as shown from the perspective of the originating router.

Figure 42 Relationship Between Destination Pattern and Session Target



The address format of the session target depends on the type of voice-network dial peer:

- VoIP—IP address, hostname of the Domain Name System (DNS) server that resolves the IP address, **ras** for registration, admission, and status (RAS) if an H.323 gatekeeper resolves the IP address, or **settlement** if the settlement server resolves the IP address
- VoFR—Interface type and number and the DLCI
- VoATM—Interface number, and ATM virtual circuit
- MMoIP—E-mail address

Note For inbound dial peers, the session target is ignored.

Digit Stripping on Outbound POTS Dial Peers

When a terminating router receives a voice call, it selects an outbound POTS dial peer by comparing the called number (the full E.164 telephone number) in the call information with the number configured as the destination pattern in the POTS dial peer. The access server or router then strips off the left-justified digits that match the destination pattern. If you have configured a prefix, the prefix is added to the front of the remaining digits, creating a dial string, which the router then dials. If all numbers in the destination pattern are stripped out, the user receives a dial tone.

For example, consider a voice call whose E.164 called number is 1(408) 555-2222. If you configure a destination-pattern of "1408555" and a prefix of "9," the router strips off "1408555" from the E.164 telephone number, leaving the extension number of "2222." It then appends the prefix, "9," to the front of the remaining numbers, so that the actual numbers dialed are "9, 2222." The comma in this example means that the router will pause for one second between dialing the "9" and dialing the "2" to allow for a secondary dial tone.

For detailed information about digit stripping and the prefix command, see the "Digit Stripping and Prefixes" section on page 187.

Configuring Dial Peers

This section describes how to configure dial peers:

Note The example configurations in this section show VoIP dial peers; the same concepts also apply to VoFR and VoATM dial peers.

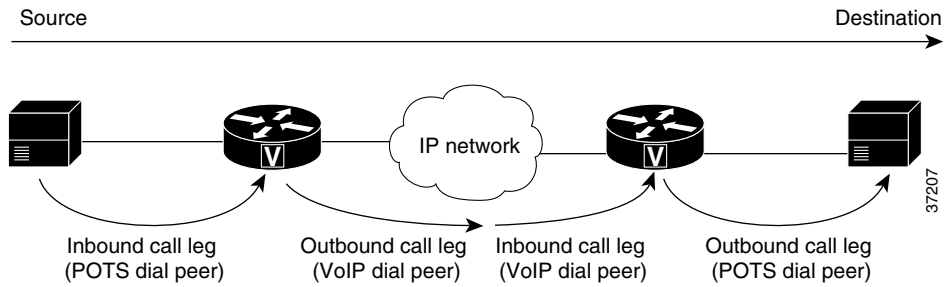
Establishing voice communication over a packet network is similar to configuring a static route: you are establishing a specific voice connection between two defined endpoints. Call legs define the discrete segments that lie between two points in the call connection. A voice call over the packet network comprises four call legs, two on the originating router and two on the terminating router; a dial peer is associated with each of these four call legs.

Configuring Dial Peers for Call Legs

When a voice call comes into the router, the router must match dial peers to route the call. For inbound calls from a POTS interface that are being sent over the packet network, the router matches a POTS dial peer for the inbound call leg and a voice-network dial peer for the outbound call leg. For calls coming into the router from the packet network, the router matches an outbound POTS dial peer to terminate the call and an inbound voice-network dial peer for features such as codec, VAD, and QoS.

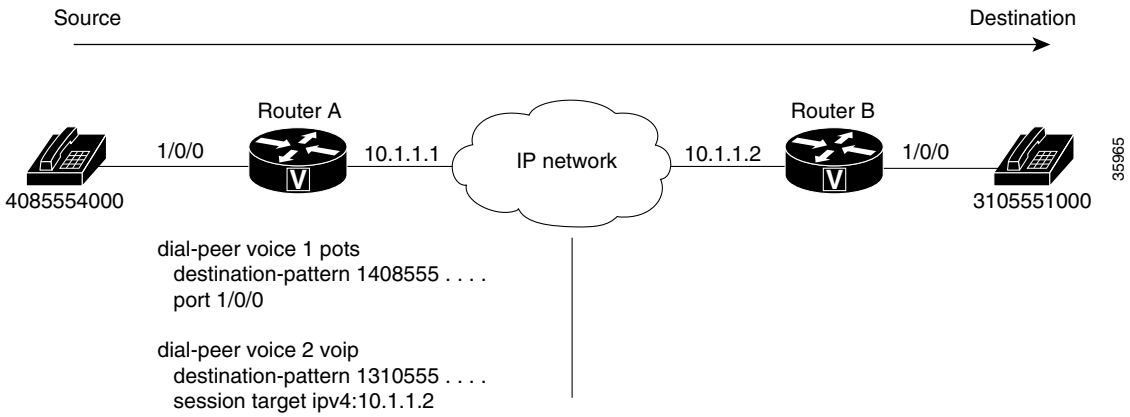
Figure 43 shows the call legs and associated dial peers necessary to complete a voice call.

Figure 43 Matching Call Legs to Dial Peers



The following configurations show an example of a call being made from 4085554000 to 3105551000. Figure 44 shows the inbound POTS dial peer and the outbound VoIP dial peer that are configured on the originating router. The POTS dial peer establishes the source of the call (via the calling number or voice port), and the voice-network dial peer establishes the destination by associating the dialed number with the network address of the remote router.

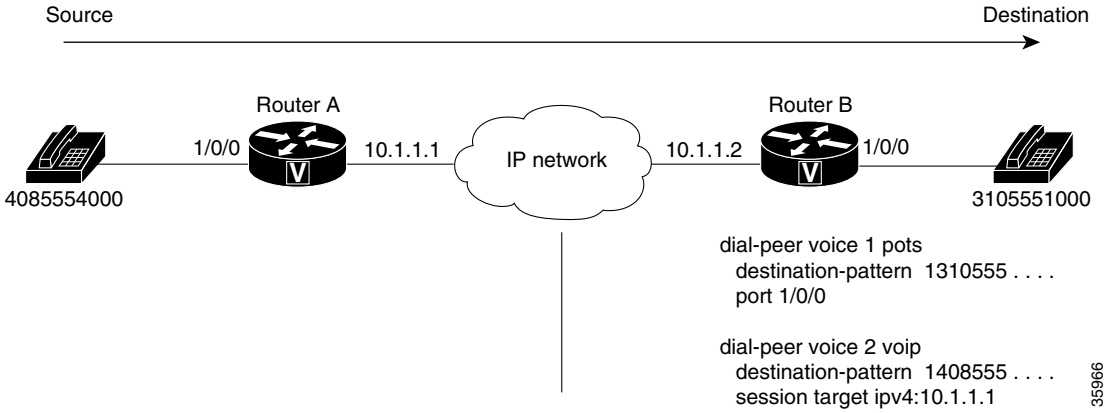
Figure 44 Dial Peers from the Perspective of the Originating Router



In this example, the dial string 14085554000 maps to telephone number 555-4000, with the digit 1 plus the area code 408 preceding the number. When you configure the destination pattern, set the string to match the local dialing conventions.

Figure 45 shows the inbound VoIP dial peer and outbound POTS dial peer that are configured on the terminating router to complete the call. Dial peers are of local significance only.

Figure 45 Dial Peers from the Perspective of the Terminating Router

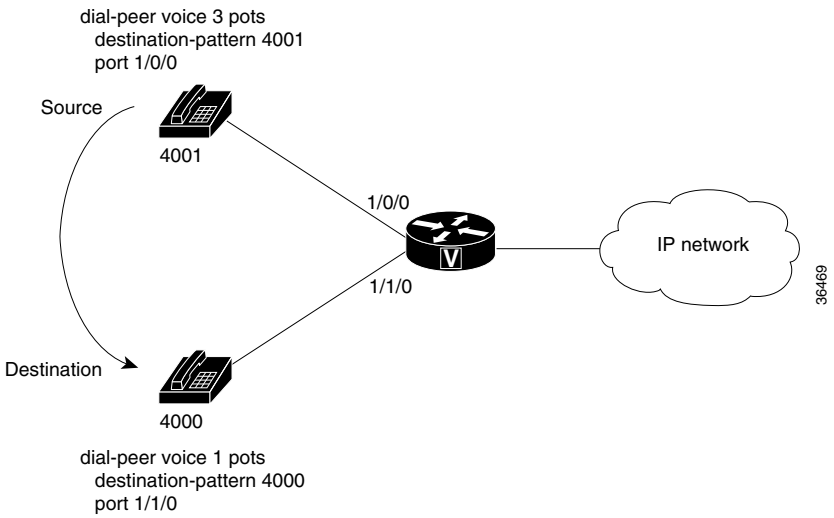


In the previous configuration examples, the last four digits in the VoIP dial peer's destination pattern were replaced with wildcards. This means that from Router A, calling any telephone number that begins with the digits "1310555" will result in a connection to Router B. This implies that Router B services all numbers beginning with those digits. From Router B, calling any telephone number that begins with the digits "1408555" will result in a connection to Router A. This implies that Router A services all numbers beginning with those digits.

Note It is not always necessary to configure the inbound dial peers. If the router is unable to match a configured dial peer for the inbound call leg, it uses an internally defined default POTS or voice-network dial peer to match inbound voice calls. In the example shown in Figure 45, dial peer 2 is only required when making a call from Router B to Router A.

The only exception to the previous example occurs when both POTS dial peers share the same router, as shown in Figure 46. In this circumstance, you do not need to configure a voice-network dial peer.

Figure 46 Communication Between Dial Peers Sharing the Same Router



This type of configuration is similar to the configuration used for hairpinning, which occurs when a voice call destined for the packet network is instead routed back over the PSTN because the packet network is unavailable. For more information about the hairpinning feature, see the "Hunt Groups and Preferences" section on page 180.

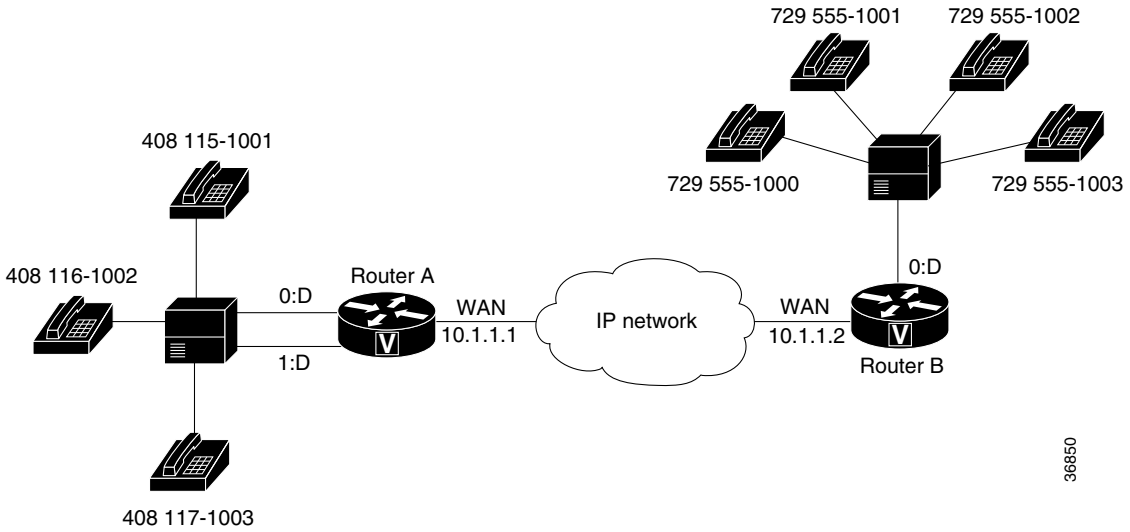
Creating a Dial Peer Configuration Table

Before you can configure dial peers, you must obtain specific information about your network. One way to identify this information is to create a dial peer configuration table. This table should contain all the telephone numbers and access codes for each router that is carrying telephone traffic in the network. Because most installations require integrating equipment into an existing voice network, the telephone dial plans are usually preset.

Figure 47 shows an example of a network in which Router A, with an IP address of 10.1.1.1, connects a small sales branch office to the main office through Router B, with an IP address of 10.1.1.2.

Note The example in Figure 47 shows a VoIP configuration. The same concepts also apply to VoFR and VoATM applications. The only change is in the format of the session target.

Figure 47 Sample VoIP Network



There are three telephone numbers in the sales branch office that need dial peers configured for them. Router B is the primary gateway to the main office; as such, it needs to be connected to the company's PBX. There are four devices that need dial peers configured for them in the main office, all of which are connected to the PBX.

Table 16 shows the peer configuration table for the example in Figure 47.

Table 16 Dial Peer Configuration Table for Sample Voice over IP Network

Dial Peer	Extension	Prefix	Destination Pattern	Type	Voice Port	Session Target
Router A						
1	51001	5	1408115. . . .	POTS	0:D	—
2	61002	6	1408116. . . .	POTS	0:D	—
3	71003	7	1408117. . . .	POTS	0:D	—
10	—	—	1408.	VoIP	—	10.1.1.2
Router B						
1	1000, 1001, 1002, 1003	—	1729555. . . .	POTS	0:D	—
10	—	—	1729.	VoIP	—	10.1.1.1

Configuring POTS Dial Peers

To configure a POTS dial peer, you must do the following:

- Identify the dial peer by assigning it a unique tag number
- Define its destination telephone number or range of telephone numbers
- Associate it with a voice port through which calls are established

Under most circumstances, the default values for the remaining dial peer configuration commands are sufficient to establish connections.

To configure a POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number pots`	Enters dial-peer configuration mode and defines a local dial peer that connects to a POTS interface. The *number* argument is one or more digits identifying the dial peer. Valid entries are from 1 to 2147483647. The **pots** keyword indicates a dial peer using basic telephone service.
Step 2	`Router(config-dial-peer)# destination-pattern string [T]`	Matches dialed digits to a telephony device. The *string* argument is a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the numbers 0 through 9 and the letters A through D. You can also enter the following special characters: • The asterisk (*) or pound sign (#) on standard touch-tone dial pads can be used anywhere in the pattern. They can be the leading character (for example, *650), except on the Cisco 3600 series. • The period (.) acts as a wildcard character. For a list of additional wildcard characters, see Table 14 on page 157. When the timer (T) character is included at the end of the destination pattern, the router collects dialed digits until the interdigit timer expires (10 seconds, by default) or until you dial the termination character (the default is #). The timer character must be a capital T.
Step 3	`Router(config-dial-peer)# port location`	Maps the dial peer to a specific logical interface. The **port** command syntax is platform-specific. For more information about the syntax of this command, see the chapter "Configuring Voice Ports" in this document.

Configuring Dial Plan Options for POTS Dial Peers

When you configure a dial plan, you have different options, depending on how the dial plan is designed. To configure optional dial plan features for POTS dial peers, use one or more of the following commands in dial-peer configuration mode:

Command	Purpose
`Router(config-dial-peer)# answer-address string`	(Optional) Selects the inbound dial peer based on the calling number.
`Router(config-dial-peer)# incoming called-number string`	(Optional) Selects the inbound dial peer based on the called number to identify voice and modem calls.
`Router(config-dial-peer)# direct-inward-dial string`	(Optional) Enables the Direct Inward Dialing (DID) call treatment for the incoming called number. For more information, see the "DID for POTS Dial Peers" section on page 178.
`Router(config-dial-peer)# forward-digits {num-digit | all | extra}`	(Optional) Configures the digit-forwarding method used by the dial peer. The valid range for the number of digits forwarded (*num-digit*) is 0 through 32. For more information, see the "Forward Digits" section on page 190.
`Router(config-dial-peer)# max-conn number`	(Optional) Specifies the maximum number of allowed connections to and from the POTS dial peer. The valid range is 1 through 2147483647.
`Router(config-dial-peer)# numbering-type {abbreviated | international | national | network | reserved | subscriber | unknown}`	(Optional) Specifies the numbering type to match, as defined by the ITU Q.931 specification. For more information, see the "Numbering Type Matching" section on page 183.
`Router(config-dial-peer)# preference value`	(Optional) Configures a preference for the POTS dial peer. The valid range is 0 through 10, where the lower the number, the higher the preference. For more information, see the "Hunt Groups and Preferences" section on page 180.
`Router(config-dial-peer)# prefix string`	(Optional) Includes a prefix that the system adds automatically to the front of the dial string before passing it to the telephony interface. Valid entries for the *string* argument are 0 through 9 and a comma (,). Use a comma to include a one-second pause between digits to allow for a secondary dial tone. For more information, see the "Digit Stripping and Prefixes" section on page 187.
`Router(config-dial-peer)# translate-outgoing {called | calling} name-tag`	(Optional) Specifies the translation rule set to apply to the calling number or called number. For more information, see the "Digit Translation Rules for VoIP" section on page 193.

Configuring VoIP Dial Peers

VoIP dial peers enable the router to make outbound calls to a particular telephony device. To configure a VoIP dial peer, you must do the following:

- Identify the dial peer by assigning it a unique tag number
- Define its destination telephone number
- Define its destination IP address

As with POTS dial peers, under most circumstances the default values for the remaining dial peer configuration commands are adequate to establish connections.

To configure a VoIP peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters dial-peer configuration mode and defines a remote VoIP dial peer. The *number* argument is one or more digits identifying the dial peer. Valid entries are from 1 to 2147483647. The **voip** keyword indicates a dial peer using voice encapsulation on the IP network.
Step 2	`Router(config-dial-peer)# destination-pattern` *string* `[T]`	Configures the dial peer's destination pattern so that the system can reconcile dialed digits with a telephone number. The *string* argument is a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the numbers 0 through 9 and the letters A through D. You can also enter the following special characters: • The asterisk (*) or pound sign (#) on standard touch-tone dial pads can be used anywhere in the pattern. They can be the leading character (for example, *650), except on the Cisco 3600 series. • The period (.) acts as a wildcard character. For a list of additional wildcard characters, see Table 14 on page 157. When the timer (T) character is included at the end of the destination pattern, the router collects dialed digits until the interdigit timer expires (10 seconds, by default) or until you dial the termination character (the default is #). The timer character must be a capital T.

	Command	Purpose
Step 3	`Router(config-dial-peer)# session target {ipv4:destination-address | dns:[$s$. | $d$. | $e$. | $u$.] host-name}`	Defines the IP address of the router that is connected to the remote telephony device. The **ipv4:***destination-address* keyword and argument indicate the IP address of the remote router. The **dns:***host-name* keyword and argument indicate that the domain name server will resolve the name of the IP address. Valid entries for this parameter are characters representing the name of the host device. Wildcards are also available for defining domain names with the keyword by using source, destination, and dialed information in the host name.
Step 4	`Router(config-dialpeer)# codec {g711alaw | g711ulaw | g723ar53 | g723ar63 | g723r53 | g723r63 | g726r16 | g726r24 | g726r32 | g728 | g729br8 | g729r8 [pre-ietf]} [bytes]`	Defines the codec for the dial peer. The optional *bytes* parameter sets the number of voice data bytes per frame. Acceptable values are from 10 to 240 in increments of 10 (for example, 10, 20, 30, and so on). Any other value is rounded down (for example, from 236 to 230). The same codec value must be configured in both VoIP dial peers on either side of the connection. If you specify **g729r8**, then IETF bit-ordering is used. For interoperability with a Cisco 2600 series, Cisco 3600 series, or Cisco AS5300 running a release earlier than Cisco IOS Release 12.0(5)T or 12.0(4)XH, you *must* specify the additional keyword **pre-ietf** after **g729r8**. The **codec** command syntax is platform- and release-specific. For more information about the syntax of this command, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

If you have used the **codec complexity** voice-card interface configuration command, the **codec** command sets the codec options that are available. If you do not set codec complexity, **g729r8** with IETF bit-ordering is used. For more information about the **codec complexity** command, see the "Configuring Voice Ports" chapter.

Configuring Codec Selection Order

You can create a voice class in which you define a selection order for codecs, and then apply the voice class to VoIP dial peers. The **voice class codec** global configuration command allows you to define the voice class containing the codec selection order. Then you use the **voice-class codec** dial-peer configuration command to apply the class to individual dial peers.

To configure codec selection order, perform the tasks described in the following sections:

- Creating a Voice Class to Define Codec Selection Order
- Applying Codec Selection Order to a VoIP Dial Peer

Creating a Voice Class to Define Codec Selection Order

To create a voice class to define the order of preference for selecting a codec when the router negotiates with a destination router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`router(config)# voice class codec tag`	Creates a voice class for a codec preference list. The range for the *tag* number is from 1 through 10000. The *tag* number must be unique on the router.
Step 2	`router(config-voice-class)# codec preference priority codec [bytes payload-size]`	Configures the order of preference for selecting a codec. Repeat this command to specify the preferred selection order for additional codecs, if required.

Applying Codec Selection Order to a VoIP Dial Peer

To apply voice-class codec attributes to a VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`router(config)# dial-peer voice tag voip`	Defines a VoIP dial peer and enters dial-peer configuration mode. The *tag* is a number that identifies the dial peer and must be unique on the router.
Step 2	`router(config-dialpeer)# voice-class codec tag`	Assigns to the dial peer the voice class that you created in the "Creating a Voice Class to Define Codec Selection Order" section. The **voice-class** command in dial-peer configuration mode is entered with a hyphen. The **voice class** command in global configuration mode is entered without the hyphen.

Note You cannot assign voice-class codec attributes to POTS dial peers.

Configuring Dial Plan Options for VoIP Dial Peers

When you configure a dial plan, you have different options, depending on how the dial plan is designed. To configure optional dial plan features, use the following commands in dial-peer configuration mode:

	Command	Purpose
Step 1	`Router(config-dial-peer)# answer-address string`	(Optional) Selects the inbound dial peer based on the calling number.
Step 2	`Router(config-dial-peer)# incoming called-number string`	(Optional) Selects the inbound dial peer based on the called number to identify voice and modem calls.

Step 3	`Router(config-dial-peer)# dtmf-relay [cisco-rtp] [h245-signal] [h245-alphanumeric]`	(Optional) Configures the tone that sounds in response to a keypress on a touch-tone telephone. Dual tone multifrequency (DTMF) tones are compressed at one end of a call and decompressed at the other end. If a low-bandwidth codec such as G.729 or G.723 is used, the tones can sound distorted. The **dtmf-relay** command transports DTMF tones generated after call establishment out-of-band by using a method that sends with greater fidelity than is possible in-band for most low-bandwidth codecs. Without DTMF Relay, calls established with low-bandwidth codecs can have trouble accessing automated telephone menu systems such as voice mail and interactive voice response (IVR) systems. A signaling method is supplied only if the remote end supports it. Options are Cisco proprietary (**cisco-rtp**), standard H.323 (**h245-alphanumeric**), and H.323 standard with signal duration (**h245-signal**).
Step 4	`Router(config-dial-peer)# fax rate {2400 | 4800 | 7200 | 9600 | 12000 | 14400 | disable | voice}`	(Optional) Specifies the transmission speed of a fax to be sent to this dial peer. The **disable** keyword turns off fax transmission capability. The **voice** keyword, which is the default, specifies the highest possible transmission speed supported by the voice rate.
Step 5	`Router(config-dial-peer)# numbering-type {abbreviated | international | national | network | reserved | subscriber | unknown}`	(Optional) Specifies the numbering type to match, as defined by the ITU Q.931 specification. For more information, see the "Numbering Type Matching" section on page 183.
Step 6	`Router(config-dial-peer)# playout-delay mode {adaptive | fixed}`	(Optional) Specifies the type of jitter buffer playout delay to use.
Step 7	`Router(config-dial-peer)# playout-delay {maximum` *value* `| nominal` *value* `| minimum {default | low | high}}`	(Optional) Specifies the amount of time that a packet is held in the jitter buffer before it is played out on the audio path. For detailed information, see the chapter "Quality of Service" in this document.
Step 8	`Router(config-dial-peer)# preference` *value*	(Optional) Configures a preference for the VoIP dial peer. The value is a number from 0 through 10, where the lower the number, the higher the preference. For more information, see the "Hunt Groups and Preferences" section on page 180.
Step 9	`Router(config-dial-peer)# tech-prefix` *number*	(Optional) Specifies that a particular technology prefix be prepended to the destination pattern of this dial peer.

Step 10	`Router(config-dial-peer)# translate-outgoing {called | calling} name-tag`	(Optional) Specifies the translation rule set to apply to the calling number or called number. For more information, see the "Digit Translation Rules for VoIP" section on page 193.
Step 11	`Router(config-dial-peer)# vad`	(Optional) Enables voice activity detection (VAD) by disabling the transmission of packets during periods of silence. VAD is enabled by default. The minimum silence detection time for VAD can be modified by using the **voice vad-time** global configuration command.

The default for the **vad** command is enabled, which is normally the preferred configuration. If you are operating on a high-bandwidth network and voice quality is of the highest importance, you should disable VAD by using the **no vad** command. This results in better voice quality, but also requires higher bandwidth for voice. For example, a broad industry average for VAD savings on links T1 and up is from 30 to 35 percent of the overall bandwidth.

Note The music threshold that is configured by using the **music-threshold** voice-port command can affect VAD performance.

Some codecs come with built-in VAD algorithms (specifically, G.729 Annex B and G.723.1 symmetric). VAD can be used with all other codecs.

Configuring VoFR Dial Peers

To configure VoFR dial peers, see the "Configuring Voice over Frame Relay" chapter.

Configuring VoATM Dial Peers

To configure VoATM dial peers, see the "Configuring Voice over ATM" chapter.

Verifying POTS and VoIP Dial Peer Configurations

You can check the validity of your dial peer configuration by performing the following tasks:

- If you have relatively few dial peers configured, you can use the **show dial-peer voice** command to verify that the configuration is correct. To display a specific dial peer or to display all configured dial peers, use this command. The following is sample output from the **show dial-peer voice** command for a specific VoIP dial peer:

```
router# show dial-peer voice 10

VoiceOverIpPeer10
        tag = 10, dest-pat = \Q',
        incall-number = \Q+14087',
        group = 0, Admin state is up, Operation state is down
        Permission is Answer,
        type = voip, session-target = \Q',
        sess-proto = cisco, req-qos = bestEffort,
        acc-qos = bestEffort,
```

```
        fax-rate = voice, codec = g729r8,
        Expect factor = 10,Icpif = 30, VAD = disabled, Poor QOV Trap = disabled,
        Connect Time = 0, Charged Units = 0
        Successful Calls = 0, Failed Calls = 0
        Accepted Calls = 0, Refused Calls = 0
        Last Disconnect Cause is ""
        Last Disconnect Text is ""
        Last Setup Time = 0
```

- To show the dial peer that matches a particular number (destination pattern), use the **show dialplan number** command. The following example displays the VoIP dial peer associated with the destination pattern 51234:

```
router# show dialplan number 51234

Macro Exp.: 14085551234
VoiceOverIpPeer1004
        tag = 1004, destination-pattern = \Q+1408555....',
        answer-address = \Q',
        group = 1004, Admin state is up, Operation state is up
        type = voip, session-target = \Qipv4:1.13.24.0',
        ip precedence: 0        UDP checksum = disabled
        session-protocol = cisco, req-qos = best-effort,
        acc-qos = best-effort,
        fax-rate = voice, codec = g729r8,
        Expect factor = 10, Icpif = 30,
        VAD = enabled, Poor QOV Trap = disabled
        Connect Time = 0, Charged Units = 0
        Successful Calls = 0, Failed Calls = 0
        Accepted Calls = 0, Refused Calls = 0
        Last Disconnect Cause is ""
        Last Disconnect Text is ""
        Last Setup Time = 0
Matched: +14085551234   Digits: 7
Target: ipv4:172.13.24.0
```

Troubleshooting Tips

You can troubleshoot your dial peer configurations by performing the following tasks:

- Ping the associated IP address to confirm connectivity. If you cannot successfully ping your destination, refer to the *Cisco IOS IP Configuration Guide*.
- To verify that the operational status and administrative status of the dial peer is up, use the **show dial-peer voice** command.

Note To activate a dial peer, the **answer-address**, **incoming called-number**, or **destination-pattern** with **port** or **session-target** command must be configured in the dial peer.

- To verify that the data is configured correctly on both routers, use the **show dialplan number** command on the local and remote routers.
- If you have configured number expansion, use the **show num-exp** command to check that the partial number on the local router maps to the correct full E.164 telephone number on the remote router.
- If you have configured translation rules, use the **test translation-rule** command to verify digit manipulation.

- If you have configured a codec value, make sure that the same codec value is configured in both VoIP dial peers on either side of the connection. You can verify the configured codec value by using the **show dial-peer voice** or **show dialplan number** command.
- To verify that the output string the router dials is correct, use the **debug voip ccapi inout** command.
- To check Real-Time Transport Protocol (RTP) packets, use the **debug cch323 rtp** command.
- To check logical channel negotiation, use the **debug cch323 h245** command.
- To check the call setup, use the **debug cch323 h225** command.

Dial Peer Overview

Before setting up a dial plan, you should understand how the router matches dialed strings to inbound and outbound dial peers. How the router matches dialed strings directly affects the digits that your users have to dial, in addition to the digits that are collected and then forwarded or played out to the telephony interface, such as a PBX, key system, or PSTN.

The following sections describe basic concepts on how the router selects a matching dial peer:

Note Unless otherwise noted, the concepts described in this section apply to VoIP, VoFR, and VoATM dial peers.

Two-Stage Dialing

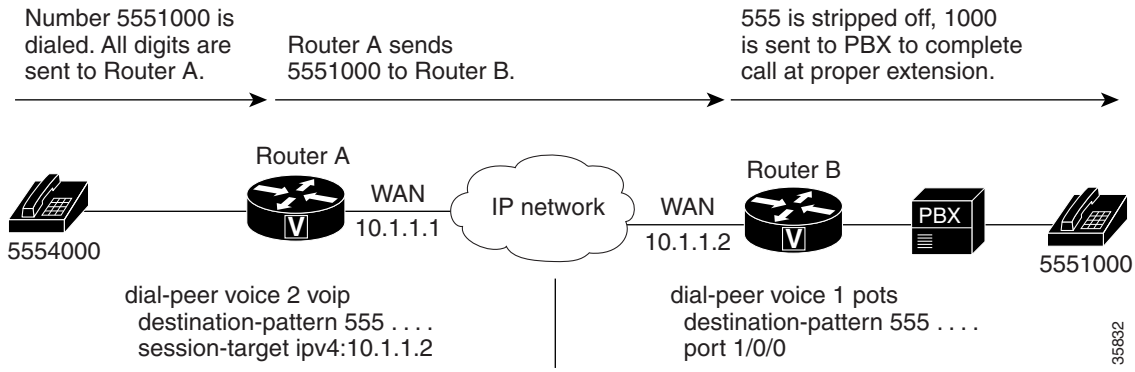
With two-stage dialing, when a voice call enters the network, the originating router collects dialed digits until it can match an outbound dial peer. As soon as the router matches a dial peer, it immediately places the call and forwards the associated dial string. No additional dialed digits are collected. The digits and wildcards that are defined in the destination pattern determine how many digits the originating router collects before matching the dial peer. Any digits dialed after the first dial peer is matched are dropped.

For example, if the dialed string is “1234599” and the originating router matches a dial peer with a destination pattern of 123. ., then the digits “99” are not collected. The call is placed immediately after the digit “5” is dialed, and the dial string “12345” is forwarded to the next call leg.

On the terminating router, the left-justified digits that explicitly match the terminating POTS dial peer are stripped off. Any trailing wildcard digits are considered excess digits. The terminating router forwards these excess digits to the telephony interface. For example, if the dial string “1234599” is matched on a terminating router to a destination pattern of “123. .,” the digits “4599” are excess digits and are forwarded to the telephony interface.

Figure 48 illustrates how the originating router collects a dial string and the terminating router forwards the digits to the telephony device.

Figure 48 Collecting and Forwarding Dialed Digits



The examples in Table 17 demonstrate how the originating router collects dialed digits for a given destination pattern in the outbound voice-network dial peer.

Table 17 Digit Collection Based on Destination Pattern

Dialed Digits	Destination Pattern	Dial String Collected[1]
5551234	5.	5551234
5551234	555. . . .	5551234
5551234	555	555
555123499	555. . . .	5551234

1. These examples apply only to two-stage dialing, in which the router collects the dialed string digit by digit. If DID is enabled in the inbound POTS dial peer, the router performs one-stage dialing, which means that the full dialed string is used regardless of the destination pattern that is matched.

The router defaults to two-stage dialing unless you configure DID. For information on configuring DID, see the "DID for POTS Dial Peers" section on page 178.

Variable-Length Matching

When matching dial peers, the router defaults to variable-length matching, which means that as long as the left-justified digits in the dial string match the configured pattern in the dial peer, any digits beyond the configured pattern are ignored for the purposes of matching. For example, dial string 5551212 would match both of the following dial peers:

```
dial-peer voice 1 voip
 destination-pattern 555
 session target ipv4:10.10.1.1

dial-peer voice 2 voip
 destination-pattern 5551212
 session target ipv4:10.10.1.2
```

To disable variable-length matching for a dial peer, add the dollar sign ($) to the end of the destination pattern, as shown:

```
dial-peer voice 1 voip
 destination-pattern 555$
 session target ipv4:10.10.1.1
```

The $ character in the above configuration prevents this dial peer from being matched for dial string 5551212 because the extra digits beyond 555 are considered in the matching.

With two-stage dialing, the router collects the dialed string digit by digit. It attempts to match a dial peer after each digit is received. As soon as it finds a match, it immediately routes the call. For example, given the following configurations, the router would immediately match dial string 5551212 to dial peer 1.

```
dial-peer voice 1 voip
 destination-pattern 555
 session target ipv4:10.10.1.1

dial-peer voice 2 voip
 destination-pattern 5551212
 session target ipv4:10.10.1.2
```

If the router is performing two-stage dialing and you want to make sure that the full dial string is collected before a dial peer is matched, you can use the timeout T-indicator as in variable-length dial plans. For example, after the router waits until the full dial string is collected, dial string 5551212 would match both of the following dial peers:

```
dial-peer voice 1 voip
 destination-pattern 555T
 session target ipv4:10.10.1.1

dial-peer voice 2 voip
 destination-pattern 5551212T
 session target ipv4:10.10.1.2
```

How the router selects a dial peer also depends on whether the dial peer is being matched for the inbound or outbound call leg. For more information, see the "Matching Inbound Dial Peers" section on page 175 and the "Matching Outbound Dial Peers" section on page 176.

Matching Inbound Dial Peers

To match inbound call legs to dial peers, the router uses three information elements in the call setup message and four configurable dial peer attributes. The three call setup elements are:

- Called number or dialed number identification service (DNIS)—A set of numbers representing the destination, which is derived from the ISDN setup message or CAS DNIS.
- Calling number or automatic number identification (ANI)—A set of numbers representing the origin, which is derived from the ISDN setup message or CAS ANI.
- Voice port—The voice port carrying the call.

The four configurable dial peer attributes are:

- Incoming called-number—A string representing the called number or DNIS. It is configured by using the **incoming called-number** dial-peer configuration command in POTS or MMoIP dial peers. For more information, see the "Identifying Voice and Modem Calls" section on page 180.
- Answer address—A string representing the calling number or ANI. It is configured by using the **answer-address** dial-peer configuration command in POTS or VoIP dial peers and is used only for inbound calls from the IP network. For more information, see the "Answer Address for VoIP" section on page 178.

- Destination pattern—A string representing the calling number or ANI. It is configured by using the **destination-pattern** dial-peer configuration command in POTS or voice-network dial peers. For more information, see the "Destination Pattern" section on page 156.
- Port—The voice port through which calls to this dial peer are placed.

The router selects an inbound dial peer by matching the information elements in the setup message with the dial peer attributes. The router attempts to match these items in the following order:

1. Called number with **incoming called-number**
2. Calling number with **answer-address**
3. Calling number with **destination-pattern**
4. Incoming voice port with configured voice port

The router must match only one of these conditions. It is not necessary for all the attributes to be configured in the dial peer or that every attribute match the call setup information; only one condition must be met for the router to select a dial peer. The router stops searching as soon as one dial peer is matched and the call is routed according to the configured dial peer attributes. Even if there are other dial peers that would match, only the first match is used.

Note For a dial peer to be matched, its administrative state must be up. The dial peer administrative state is up by default when it is configured with at least one of these commands: **incoming called-number**, **answer-address**, or **destination-pattern**. If **destination-pattern** is used, the voice port or session target must also be configured.

Inbound Dial Peers for IVR Applications

To identify an interactive voice response (IVR) application to handle inbound calls, the originating router must match a POTS dial peer. You configure which IVR application handles incoming voice calls by using the **application** dial-peer configuration command. If the router is unable to match an inbound dial peer, or if the inbound dial peer does not specify an application, the default application handles the call. The following configuration shows an example of specifying an IVR application for an inbound POTS call leg:

```
dial-peer voice 571 pots
 application tr6
 destination-pattern 5714954
 port 0:D
```

Matching Outbound Dial Peers

How the router selects an outbound dial peer depends on whether DID is configured in the inbound POTS dial peer. If DID is not configured in the inbound POTS dial peer, the router collects the incoming dialed string digit by digit. As soon as one dial peer is matched, the router immediately places the call using the configured attributes in the matching dial peer.

If DID is configured in the inbound POTS dial peer, the router uses the full incoming dial string to match the destination pattern in the outbound dial peer. With DID, the setup message contains all the digits necessary to route the call; no additional digit collection is required. If more than one dial peer matches the dial string, all of the matching dial peers are used to form a rotary group. The router attempts to place the outbound call leg using all of the dial peers in the rotary group until one is successful. For more information on rotary groups, see the "Hunt Groups and Preferences" section on page 180.

For information on configuring DID, see the "DID for POTS Dial Peers" section on page 178.

Default Routes for Outbound Call Legs

Default routes reduce the number of dial peers that must be configured when calls that are not terminated by other dial peers are sent to a central router, usually for forwarding to a PBX. A default route is a dial peer that automatically matches any call that is not terminated by other dial peers. For example, in the following configuration, the destination pattern 8. . . is a voice default route because all voice calls with a dialed string that starts with 8 followed by at least three additional digits will either match on 8208 or end up with 8. . ., which is the last-resort voice route used by the router if no other dial peer is matched.

```
dial-peer voice 8 pots
 destination-pattern 8208
 port 1/1
!
dial-peer voice 1000 pots
 destination-pattern 8...
 port 1/1
```

A default route could also be defined by using a single wildcard character with the timeout T-indicator in the destination pattern, as shown in the following example:

```
dial-peer voice 1000 voip
 destination-pattern .T
 session-target ipv4:10.10.1.2
```

You should be careful, however, when using the T-indicator for default routes. Remember, when matching dial peers for outbound call legs, the router places the call as soon as it finds the first matching dial peer. The router could match on this dial peer immediately even if there were another dial peer with a more explicit match and a more desirable route.

Note The timeout T-indicator is appropriate only for two-stage dialing. If the router is configured for one-stage dialing, which means that DID is configured in the inbound POTS dial peer, then the timeout T-indicator is unnecessary.

Configuring Dial Peer Matching Features

You can define the attributes that the router uses to match dial peers by configuring specific dial peer features. These dial peer matching features are described in the following sections:

- Numbering Type Matching, page 183
- Class of Restrictions, page 184

Note Unless otherwise noted, the concepts described in this section apply to VoIP, VoFR, and VoATM dial peers.

Answer Address for VoIP

The **answer-address** command can be used to select the inbound dial peer for VoIP calls, instead of using the destination pattern. If the **answer-address** command is configured in VoIP or POTS dial peers, the router attempts to match the calling number to the string configured as the answer address before attempting to match a destination pattern in any dial peer. The following dial peer would match any inbound VoIP call that had a calling number of 5551212.

```
dial-peer voice 2 voip
 answer-address 5551212
 session target ipv4:192.168.1.1
```

For more information, see the "Matching Inbound Dial Peers" section on page 175.

Note The **answer-address** command is not supported for VoFR or VoATM dial peers.

DID for POTS Dial Peers

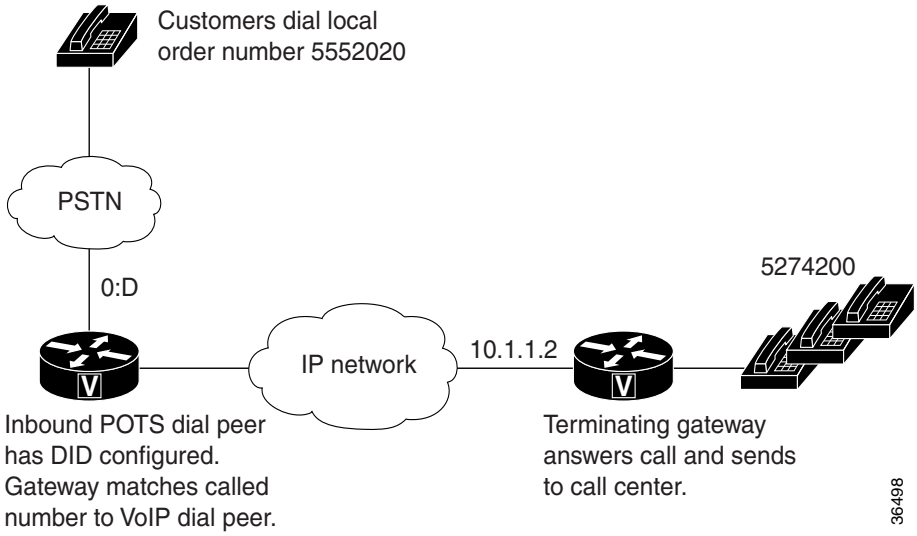
The Direct Inward Dialing (DID) feature in dial peers enables the router to use the called number (DNIS) to directly match an outbound dial peer when receiving an inbound call from a POTS interface. When DID is configured on the inbound POTS dial peer, the called number (DNIS) is automatically used to match the destination pattern for the outbound call leg.

Unless otherwise configured, when a voice call comes into the router, the router presents a dial tone to the caller and collects digits until it can identify an outbound dial peer. This process is called *two-stage dialing*. After the outbound dial peer is identified, the router forwards the call through to the destination as configured in the dial peer.

You may prefer that the router use the called number (DNIS) to find a dial peer for the outbound call leg—for example, if the switch connecting the call to the router has already collected all the dialed digits. DID enables the router to match the called number to a dial peer and then directly place the outbound call. With DID, the router does not present a dial tone to the caller and does not collect digits; it forwards the call directly to the configured destination. This is called *one-stage dialing*.

Figure 49 shows a call scenario using DID.

Figure 49 VoIP Call Using DID



In Figure 49, the POTS dial peer that matches the incoming called-number has direct-inward-dial configured:

```
dial-peer voice 100 pots
 incoming called-number 5552020
 direct-inward-dial
 port 0:D
```

The **direct-inward-dial** command in the POTS dial peer tells the gateway to look for a destination pattern in a dial peer that matches the DNIS. For example, if the dialed number is 5552020, the gateway matches the following VoIP dial peer for the outbound call leg:

```
dial-peer voice 101 voip
 destination-pattern 5552020
 session target ipv4:10.1.1.2
```

The call is made across the IP network to 10.1.1.2, and a match is found in that terminating gateway:

```
dial-peer voice 555 pots
 destination-pattern 5552020
 port 0:D
 prefix 5274200
```

This dial peer matches on the dialed number and changes that number to 52744200 with the **prefix** command. The result is that the user dials a number, gets connected, and never knows that the number reached is different from the number dialed.

Note DID for POTS dial peers is not the same as analog DID for Cisco routers which enables DID trunk service from the PSTN.

To configure a POTS dial peer for DID, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode and defines a local dial peer that will connect to the POTS network. The *number* is one or more digits identifying the dial peer. Valid entries are from 1 to 2147483647.
Step 2	`Router(config-dial-peer)# direct-inward-dial`	Specifies DID for this POTS dial peer.

Note DID is configured for inbound POTS dial peers only.

Identifying Voice and Modem Calls

When a Cisco router is handling both modem and voice calls, it needs to identify the service type of the call—that is, whether the incoming call to the router is a modem or a voice call. When the router handles only modem calls, the service type identification is handled through modem pools. Modem pools associate calls with modem resources based on the called number (DNIS). In a mixed environment, where the router receives both modem and voice calls, you need to identify the service type of a call by using the **incoming called-number** command.

If the **incoming called-number** command is not configured, the router attempts to resolve whether an incoming call is a modem or voice call on the basis of the interface over which the call comes. If the call comes in over an interface associated with a modem pool, the call is assumed to be a modem call; if a call comes in over a voice port associated with a POTS dial peer, the call is assumed to be a voice call.

To identify the service type of a call as voice, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* {**pots** | **voip** | **vofr** | **voatm**}	Enters dial peer configuration mode.
Step 2	`Router(config-dial-peer)# incoming called-number` *number*	Defines the telephone number that identifies voice calls associated with this dial peer.

Hunt Groups and Preferences

The router supports the concept of *hunt groups*, sometimes called *rotary groups*, in which multiple dial peers are configured with the same destination pattern. Because the destination of each POTS dial peer is a single voice port to a telephony interface, hunt groups help ensure that calls get through even when a specific voice port is busy. If the router is configured to hunt, it can forward a call to another voice port when one voice port is busy.

For example, in the following configuration for Router A, four POTS dial peers are configured with different destination patterns. Because each dial peer has a different destination pattern, no backup is available if the voice port mapped to a particular dial peer is busy with another call.

With a hunt group, if a voice port is busy, the router hunts for another voice port until it finds one that is available. In the following example for Router B, each dial peer is configured using the same destination pattern of 3000, forming a dial pool to that destination pattern.

Router A (Without Hunt Groups)	Router B (With Hunt Groups and Preferences)
dial-peer voice 1 pots destination-pattern 3001 port 1/1 ! dial-peer voice 2 pots destination-pattern 3002 port 1/2 ! dial-peer voice 3 pots destination-pattern 3003 port 1/3 ! dial-peer voice 4 pots destination-pattern 3004 port 1/4	dial-peer voice 1 pots destination pattern 3000 port 1/1 preference 0 ! dial-peer voice 2 pots destination pattern 3000 port 1/2 preference 1 ! dial-peer voice 3 pots destination pattern 3000 port 1/3 preference 2 ! dial-peer voice 4 pots destination pattern 3000 port 1/4 preference 3

To give specific dial peers in the pool a preference over other dial peers, you can configure the preference order for each dial peer by using the **preference** command. The router attempts to place a call to the dial peer with the highest preference. The configuration example given for Router B shows that all dial peers have the same destination pattern, but different preference orders.

The lower the preference number, the higher the priority. The highest priority is given to the dial peer with preference order 0. If the same preference is defined in multiple dial peers with the same destination pattern, a dial peer is selected randomly.

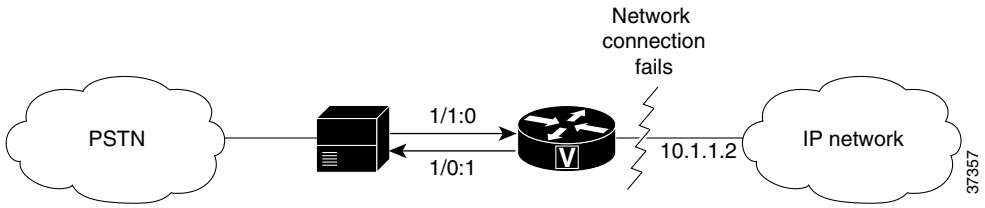
By default, dial peers in a hunt group are selected according to the following criteria, in the order listed:

1. Longest match in phone number—Destination pattern that matches the greatest number of dialed digits. For example, if one dial peer is configured with a dial string of 345. . . . and a second dial peer is configured with 3456789, the router would first select 3456789 because it has the longest explicit match of the two dial peers.
2. Explicit preference—Priority configured by using the **preference** dial peer command.
3. Random selection—All destination patterns weighted equally.

You can change this default selection order or choose different methods for hunting dial peers by using the **dial-peer hunt** global configuration command. An additional selection criteria is “least recent use,” which selects the destination pattern that has waited the longest since being selected.

You can mix POTS and voice-network dial peers when creating hunt groups. This can be useful if you want incoming calls to be sent over the packet network, except that if network connectivity fails, you want to reroute the calls back through the PBX to the PSTN. This type of configuration is sometimes referred to as *hairpinning*. Hairpinning is illustrated in Figure 50.

Figure 50 Voice Call Using Hairpinning



The following configuration shows an example of sending calls to the PSTN if the IP network fails:

```
dial-peer voice 101 voip
 destination-pattern 472....
 session target ipv4:192.168.100.1
 preference 0
!
dial-peer voice 102 pots
 destination-pattern 472....
 prefix 472
 port 1/0:1
 preference 1
```

You cannot use the same preference numbers for POTS and voice-network dial peers within a hunt group. You can set a separate preference order for each dial peer type, but the preference order does not work on both at the same time. For example, you can configure preference order 0, 1, and 2 for POTS dial peers, and you can configure preference order 0, 1, and 2 for the voice-network dial peers, but the two preference orders are separate. The system resolves preference orders among POTS dial peers first.

Configuring Dial-Peer Hunting Options

Dial-peer hunting is enabled by default. To disable dial-peer hunting on a dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* {**pots** | **vofr** | **voip**}	Enters dial-peer configuration mode for the specified dial peer.
Step 2	`Router(config-dial-peer)# huntstop`	(Optional) Disables dial-peer hunting on the dial peer. Once you enter this command, no further hunting is allowed if a call fails on the selected dial peer.

Use the **no huntsto**p command to reenable dial-peer hunting if it has been disabled.

To configure dial peer hunting options for all dial peers, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer hunt` *hunt-order-number*	(Optional) Specifies the hunt selection order for dial peers in a hunt group. Valid entries are 0 through 7. The default is 0. • 0—Longest match in phone number, explicit preference, random selection • 1—Longest match in phone number, explicit preference, least recent use • 2—Explicit preference, longest match in phone number, random selection • 3—Explicit preference, longest match in phone number, least recent use • 4—Least recent use, longest match in phone number, explicit preference • 5—Least recent use, explicit preference, longest match in phone number • 6—Random selection • 7—Least recent use
Step 2	`Router(config)# voice hunt {user-busy | invalid-number | unassigned-number}`	(Optional) Defines how the originating or tandem router handles rotary dial-peer hunting if it receives a disconnect cause code from the terminating router. • **user-busy** sets the router to continue dial-peer hunting if it receives a user-busy disconnect cause code from a destination router. • **invalid-number** sets the router to stop dial-peer hunting if it receives a an invalid-number disconnect cause code from a destination router. • **unassigned-number** sets the router to stop dial-peer hunting if it receives an unassigned-number disconnect cause code from a destination router.

Numbering Type Matching

A dial peer can be selected according to the type of number field in the called party number or calling party number information element, in addition to matching the dial peer based on the configured destination pattern, answer address, or incoming called number. The type of number value is selected by using the **numbering-type** dial-peer configuration command.

For example, in the following configuration, the dialed string “4085559999” would match this dial peer if the type of number field for the called party number is “national.”

```
dial-peer voice 408 voip
 numbering-type national
 destination-pattern 408.......
 session target ipv4:10.1.1.2
```

The following numbering types can be used:

- Abbreviated—Abbreviated representation of the complete number as supported by this network
- International—Number called to reach a subscriber in another country
- National—Number called to reach a subscriber in the same country, but outside the local network
- Network—Administrative or service number specific to the serving network
- Reserved—Reserved for extension
- Subscriber—Number called to reach a subscriber in the same local network
- Unknown—Type of number is unknown by the network

For detailed information about these numbering types, see ITU-T Recommendation Q.931

Configuring Numbering-Type Matching

To configure numbering-type matching for a dial peer call leg, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* {**pots** | **voip** | **vofr** | **voatm**}	Enters dial peer configuration mode.
Step 2	`Router(config-dial-peer)# numbering-type` {**abbreviated** | **international** | **national** | **network** | **reserved** | **subscriber** | **unknown**}	Specifies the numbering type to match, as defined by the ITU Q.931 specification.

Note To match a dial peer using the **numbering-type** command, you must also configure the **destination-pattern**, **answer-address**, or **incoming called-number** command.

Class of Restrictions

The Class of Restrictions (COR) feature provides the ability to deny certain call attempts based on the incoming and outgoing class of restrictions provisioned on the dial peers. This functionality provides flexibility in network design, allows users to block calls (for example, to 900 numbers), and applies different restrictions to call attempts from different originators.

Note COR is supported only on the Cisco AS5800 access server.

COR is used to specify which incoming dial peer can use which outgoing dial peer to make a call. Each dial peer can be provisioned with an incoming and an outgoing COR list. The incoming COR list indicates the capability of the dial peer to initiate certain classes of calls. The outgoing COR list indicates the capability required for an incoming dial peer to deliver a call via this outgoing dial peer. If the capabilities of the incoming dial peer are not the same or a superset of the capabilities required by the outgoing dial peer, the call cannot be completed using this outgoing dial peer.

A typical application of COR is to define a COR name for the number that an outgoing dial peer serves, then define a list that contains only that COR name, and assign that list as **corlist outgoing** for this outgoing dial peer. For example, dial peer with destination pattern 5T can have a **corlist outgoing** that contains COR 5x, as shown in the following configuration.

The next step, in the typical application, is to determine how many call permission groups are needed, and define a COR list for each group. For example, group A is allowed to call 5x and 6x, and group B is allowed to call 5x, 6x, and 1900x. Then, for each incoming dial peer, we can assign a group for it, which defines what number an incoming dial peer can call. Assigning a group means assigning a **corlist incoming** to this incoming dial peer.

```
dial-peer cor custom
 name 5x
 name 6x
 name 1900x
!
dial-peer cor list listA
 member 5x
 member 6x
!
dial-peer cor list listB
 member 5x
 member 6x
 member 1900x
!
dial-peer cor list list5x
 member 5x
!
dial-peer cor list list6x
 member 6x
!
dial-peer cor list list1900x
 member 1900x

! outgoing dialpeer 100, 200, 300
dial-peer voice 100 pots
 destination-pattern 5T
 corlist outgoing list5x
dial-peer voice 200 pots
 destination-pattern 6T
 corlist outgoing list6x
dial-peer voice 300 pots
 destination-pattern 1900T
 corlist outgoing list1900x
!
! incoming dialpeer 400, 500
dial-peer voice 400 pots
 answer-address 525....
 corlist incoming listA
dial-peer voice 500 pots
 answer-address 526
 corlist incoming listB
```

Configuring Classes of Restrictions

To configure classes of restrictions for dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer cor custom`	Enters COR configuration mode to specify classes of restrictions to apply to dial peers.
Step 2	`Router(config-dp-cor)# name class-name`	Provides a name for a class of restrictions. **Note** Repeat this step for additional class names, as needed. These class names are used to define the COR lists configured in Step 4 and Step 5.
Step 3	`Router(config-dp-cor)# exit`	Exits COR configuration mode.
Step 4	`Router(config)# dial-peer cor list list-name`	Provides a name for a list of restrictions.
Step 5	`Router(config-dp-corlist)# member class-name`	Adds a COR class to this list of restrictions. The member is a class named in Step 2. **Note** Repeat Step 4 and Step 5 to define another list and its membership, as needed.
Step 6	`Router(config-dp-corlist)# exit`	Exits COR-list configuration mode.
Step 7	`Router(config)# dial-peer voice number {pots | voip}`	Enters dial-peer configuration mode and defines a dial peer.
Step 8	`Router(config-dial-peer)# corlist incoming cor-list-name`	Specifies the COR list to be used when this is the incoming dial peer.
Step 9	`Router(config-dial-peer)# corlist outgoing cor-list-name`	Specifies the COR list to be used when this is the outgoing dial peer. **Note** Repeat Step 7 through Step 9 for additional dial peers, as needed.

Verifying Classes of Restrictions

To check the validity of your classes of restrictions configuration, perform the following tasks:

- Enter the **show dial-peer voice** command to learn whether the COR list fields are set as desired on a dial peer:

```
Router# show dial-peer voice 210

VoiceEncapPeer210
        information type = voice,
        tag = 210, destination-pattern = `221',
        answer-address = `', preference=0,
        numbering Type = `unknown'
        group = 210, Admin state is up, Operation state is up,
        incoming called-number = `221', connections/maximum = 4/unlimited,
        DTMF Relay = disabled,
        Modem = system passthrough ,
        huntstop = disabled,
        application associated:
        permission :both
        incoming COR list:maximum capability
        outgoing COR list:minimum requirement
```

```
        type = pots, prefix = `221',
        forward-digits default
        session-target = `', voice-port = `1/0/8:D',
        direct-inward-dial = enabled,
        digit_strip = enabled,
```

- Enter the **show dial-peer cor** command to display the COR names and lists you defined:

```
Router# show dial-peer cor

Class of Restriction
  name:900block
  name:800_call
  name:Catchall

COR list <list1>
  member:900block
  member:800_call

COR list <list2>
  member:900block

COR list <list3>
  member:900block
  member:800_call
  member:Catchall
```

Configuring Digit Manipulation

The router may need to manipulate digits in a dial string before it passes the dial string to the telephony device. This can be necessary, for instance, when calling PBXs with different capabilities to accept digits, or for PSTN and international calls. You may need to consider different strategies for configuring digit manipulation within your dial peers depending on your existing dial plan, the digits users are expected to dial, and the capabilities of your PBX or key system unit (KSU). These digit-manipulation options, in conjunction with the destination pattern, determine the dial string that the router forwards to the telephony device.

The following dial peer digit-manipulation options are described in this section:

- Digit Stripping and Prefixes, page 187
- Forward Digits, page 190
- Number Expansion, page 191
- Digit Translation Rules for VoIP, page 193

Note Unless otherwise noted, these concepts apply to VoIP, VoFR, and VoATM networks.

Digit Stripping and Prefixes

When the terminating router matches a dial string to an outbound POTS dial peer, by default the router strips off the left-justified digits that explicitly match the destination pattern. Any remaining digits, called *excess digits,* are forwarded to the telephony interface, such as a PBX or the PSTN. For more information about excess digits, see the "Two-Stage Dialing" section on page 173.

Some telephony interfaces require that any digits that are stripped from the dial string must be recovered to support a particular dial plan. You can accomplish this either by using the **no digit-strip** dial-peer configuration command to disable the default digit-stripping behavior or by using the **prefix** dial-peer configuration command to add digits to the front of the dial string before it is forwarded to the telephony interface. These commands are supported only in POTS dial peers.

The **no digit-strip** command disables the automatic digit-stripping function so that matching digits are not stripped from the dialed string before it is passed to the telephony interface. For example, in the following dial peer configuration, the entire seven-digit dialed string is passed to the telephony interface:

```
dial-peer voice 100 pots
 destination-pattern 555....
 no digit-strip
 port 1/0:1
```

Disabling digit stripping is useful when the telephony interface requires the full dialed string. With some dial plans, however, the dialed digits must be manipulated according to specific rules. The **prefix** command can be used to add specific digits to the front of the dialed string before it is forwarded to the telephony interface.

For example, consider a telephone whose E.164 called number is 1(408)555-1234. This telephone can be reached within the company by dialing its extension number, 51234. If you configure a destination pattern of "1408555. . . . " (the periods represent wildcards) for the associated outbound POTS dial peer, the terminating gateway will strip off the digits "1408555" when it receives a call for 1(408)555-1234. For the terminating gateway to forward the call to the appropriate destination, the digit "5" needs to be prepended to the remaining digits. In this case, you would configure a prefix of 5, as shown in the following dial peer configuration.

```
dial-peer voice 100 pots
 destination-pattern 1408555....
 prefix 5
 port 1/0:1
```

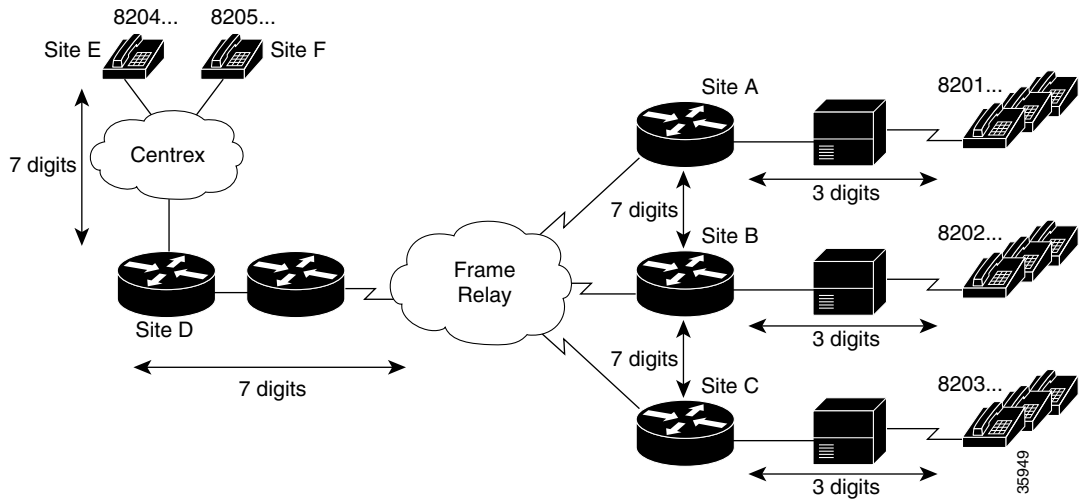
A prefix can also include commas (,). Each comma indicates a one-second pause in dialing. For example, consider a telephone whose E.164 called number is 1(408)555-1234; to reach this device, you must dial "9." In this case, you might configure "1408. . . . . " as the destination pattern, and "9" as the prefix. In this example, the terminating router will strip the digits "1408" from the called number and append the digit "9" to the front of the remaining digits, so that the actual number dialed is" 9,5551234." The router pauses for one second between dialing the "9" and the "5551234" to allow for a secondary dial tone. In this example, you would configure the router as follows:

```
dial-peer voice 100 pots
 destination-pattern 1408.......
 prefix 9,
 port 1/0:1
```

Using a comma with the **prefix** command is useful when the router must allow for a secondary dial tone; otherwise the router does not wait for the dial tone before playing out excess digits. Putting commas in the prefix makes the router pause one second per comma, allowing for a dial tone to occur before the router plays out the digits.

Figure 51 shows an example of a network using the **no digit-strip** command. In this example, a central site (Site D) is connected to remote sites through routers (Sites A, B, and C), as well as through a Centrex system for sites still using the PSTN (Sites E and F). The Centrex service requires the full 7-digit dial string to complete calls. The dial peers are configured with a fixed-length 7-digit dial plan.

Figure 51 Network with Digit Stripping Disabled or Prefixes Enabled



When Site E (8204. . .) dials 8201999, the full 7-digit dialed string is passed through the Centrex to the router at Site D. Router D matches the destination pattern 8201. . . and forwards the 7-digit dial string to Router A. Router A matches the destination pattern 8201. . . , strips off the matching 8201, and forwards the remaining 3-digit dial string to the PBX. The PBX matches the correct station and completes the call to the proper extension.

Calls in the reverse direction are handled similarly, but because the Centrex service requires the full 7-digit dial string to complete calls, the POTS dial peer at Router D is configured with digit stripping disabled. Alternatively, digit stripping could be enabled and the dial peer could instead be configured with a 4-digit prefix, in this case 8204, which would result in forwarding the full dial string to the Centrex service.

Router A	Router D
dial-peer voice 1 pots destination-pattern 8201... port 1/0:1 ! dial-peer voice 4 vofr destination-pattern 8204... session target s0 2 ! dial-peer voice 5 vofr destination-pattern 8205... session target s0 2 !	dial-peer voice 4 pots destination-pattern 8204... **no digit-strip** port 1/0:1 ! dial-peer voice 5 pots destination-pattern 8205... **no digit-strip** port 1/0:1 ! dial-peer voice 1 vofr destination-pattern 8201... session target s0 1 !

Forward Digits

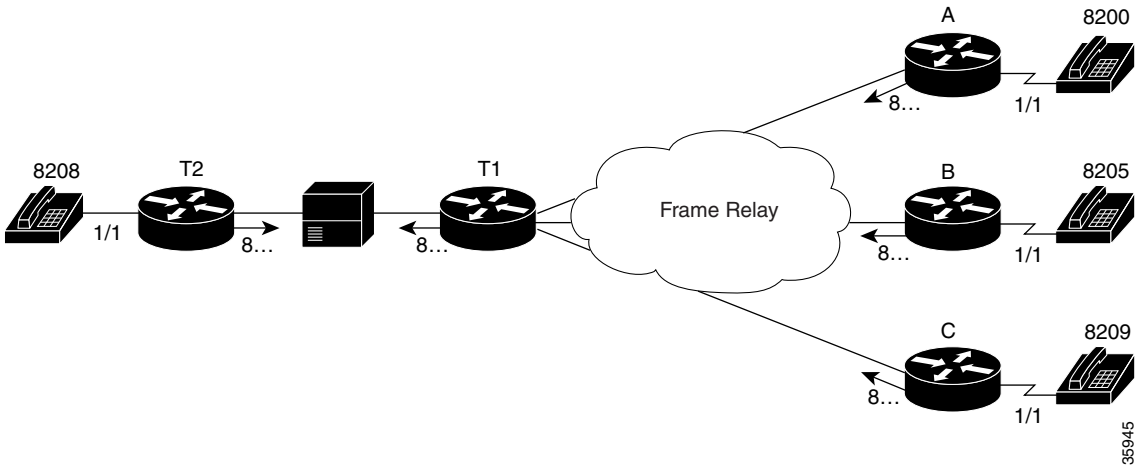
The **forward-digits** command controls the number of digits that are stripped before the dialed string is passed to the telephony interface. On outbound POTS dial peers, the terminating router normally strips off all digits that explicitly match the destination pattern in the terminating POTS dial peer. Only digits matched by the wildcard pattern are forwarded. The **forward-digits** command can be used to forward a fixed number of dialed digits, or all dialed digits, regardless of the number of digits that explicitly match the destination pattern.

For example, the **forward-digits 4** command tells the router to forward the last four digits in the dialed string. The **forward-digits all** command instructs the router to forward the full dialed string. If the length of the dialed string is longer than the length of the destination pattern, the **forward-digits extra** command forwards the extra trailing digits. Extra digits are not forwarded, however, if the dial-peer destination pattern is variable length; for example, 123T, 123 . . . T.

The **forward-digits** command is supported only in POTS dial peers.

Figure 52 shows an example of routing voice calls through a PBX using forward digits. In this configuration, Routers T1 and T2 are tandem nodes that must support forward digits so that calls from Routers A, B, or C can make a call to extension 8208.

Figure 52 Routing Voice Calls Through a PBX Using Forward Digits



In this example, all digits matched with destination 8 . . . are forwarded to the appropriate port. For a call from Router A to reach extension 8208, the call first terminates at Router T1, which plays out the digits 8208 to the voice port connected to the PBX. The PBX then routes the voice call to Router T2. The **forward-digits all** command is used here, but the **forward-digits 4** command could also be used in this example.

The following dial peer configurations are required on each router for this example:

Router T1

```
dial-peer voice 1 vofr
 destination-pattern 8200
 session-target s0 1
!
dial-peer voice 6 vofr
 destination-pattern 8205
 session-target s0 6
!
dial-peer voice 10 vofr
 destination-pattern 8209
 session-target s0 10
!
dial-peer voice 1 pots
 destination-pattern 8...
 forward-digits all
 port 1/1
```

Router T2

```
dial-peer voice 8 pots
 destination-pattern 8208
 port 1/1
!
dial-peer voice 1000 pots
 destination-pattern 8...
 forward-digits all
 port 1/1
!
dial-peer voice 9999 pots
 destination-pattern ....
 forward-digits all
 port 1/1
```

Router A

```
dial-peer voice 1 pots
 destination-pattern 8200
 port 1/1
!
dial-peer voice 1000 vofr
 destination-pattern 8...
 session-target s0 1
```

Number Expansion

In most corporate environments, the telephone network is configured so that you can reach a destination by dialing only a portion (an extension number) of the full E.164 telephone number. You can define an extension number as the destination pattern for a dial peer. The router can be configured to recognize the extension number and expand it into its full E.164 dialed number when the **num-exp** global configuration command is used with the **destination-pattern** dial-peer configuration command.

Number expansion is a globally applied rule that enables you to define a set of digits for the router to prepend to the beginning of a dialed string before passing it to the remote telephony device. This reduces the number of digits that a user must dial to reach a remote location. Number expansion is similar to using a prefix, except that number expansion is applied globally to all dial peers.

Using a simple telephony-based example, suppose that John works in a company where employees extensions are reached by dialing the last four digits of the full E.164 telephone number. The E.164 telephone number is 555-2123; John's extension number is 2123. Suppose that every employee on John's floor has a telephone number that begins with the same first four digits: 5552. You could define each dial peer's destination pattern using each extension number, and then use number expansion to prepend the first four digits onto the extension. In this example, the router could be configured as follows:

```
num-exp 2... 5552...
dial peer voice 1 pots
 destination pattern 2123
```

Number expansion can also be used to replace a dialed number with another number, as in the case of call forwarding. Suppose that for some reason, John needs to have all of his telephone calls forwarded to another number, 555-6611. In this example, you would configure the router as follows:

```
num-exp 2123 5556611
dial peer voice 1 pots
 destination pattern 2123
```

In this example, every time the device receives a call for extension 2123, the dialed digits will be replaced with 555-6611 and the call will be forwarded to that telephone.

Before you configure the **num-exp** command, it is helpful to map individual telephone extensions to their full E.164 dialed numbers. This task can be done easily by creating a number expansion table.

Creating a Number Expansion Table

Figure 53 shows a network for a small company that wants to use VoIP to integrate its telephony network with its existing IP network. The destination patterns (or expanded telephone numbers) associated with Router A are 408 115-xxxx, 408 116-xxxx, and 408 117-xxxx, where xxxx identifies the individual dial peers by extension. The destination pattern (or expanded telephone number) associated with Router B is 729 555-xxxx.

Figure 53 VoIP Example for Number Expansion

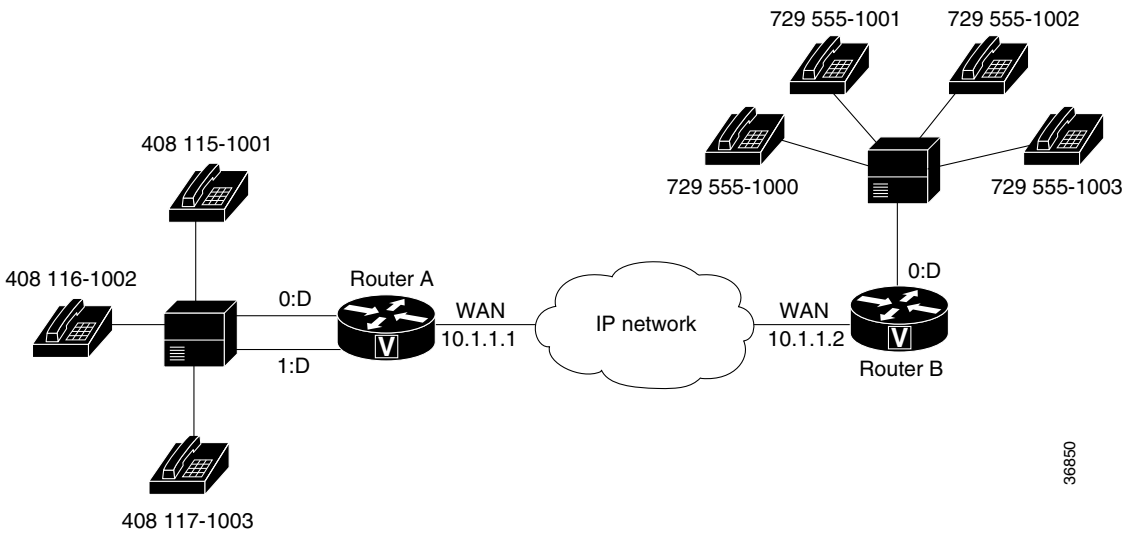


Table 18 shows the number expansion table for this scenario. The information included in this example must be configured on both Router A and Router B.

Table 18 Sample Number Expansion Table

Extension	Destination Pattern	Num-Exp Command Entry
5. . . .	408115. . . .	num-exp 5. . . . 408115. . . .
6. . . .	408116. . . .	num-exp 6. . . . 408116. . . .
7. . . .	408117. . . .	num-exp 7. . . . 408117. . . .
1. . .	729555. . . .	num-exp 1. . . 729555. . . .

The period (.) character represents wildcards (such as extension numbers) in a telephone number.

Configuring Number Expansion

To expand an extension number into its full telephone number, use the following command in global configuration mode:

Command	Purpose
`Router(config)# num-exp extension-number expanded-number`	Configures number expansion globally for all dial peers. The *extension-number* argument defines the extension number to expand into the full telephone number that is specified by the *expanded-number* argument. The *expanded-number* argument defines the full telephone number or destination pattern to which the extension number is expanded.

Verifying Number Expansion

You can check the validity of your number expansion configuration by performing the following tasks:

- Enter the **show num-exp** command to confirm that you have mapped the telephone numbers correctly.
- Enter the **show dialplan number** command to see how a telephone number maps to a dial peer.

Digit Translation Rules for VoIP

Digit translation rules are used to manipulate the calling number (ANI) or called number (DNIS) digits for a voice call, or to change the numbering type of a call. Translation rules are used to convert a telephone number into a different number before the call is matched to an inbound dial peer or before the call is forwarded by the outbound dial peer. For example, within your company you may dial a five-digit extension to reach an employee at another site. If the call is routed through the PSTN to reach the other site, the originating gateway must use translation rules to convert the five-digit extension into the 10-digit format that is recognized by the central office switch.

Translation rules are defined by using the **translation-rule** command. After you define a set of translation rules, you can apply the rules to all inbound VoIP calls, to all inbound calls that terminate at a specific voice port, and to individual inbound or outbound call legs according to the dial peer.

Note Digit translation rules are not supported for inbound SIP calls.

The following example shows a dial peer that is configured to use translation-rule set 1, which contains ten translation rules. The first rule defined is rule 0, in which 910 is the pattern that must be matched and replaced, and 0 is the pattern that is substituted for 910.

```
translation-rule 1
 rule 0 ^910 0
 rule 1 ^911 1
 rule 2 ^912 2
 rule 3 ^913 3
 rule 4 ^914 4
 rule 5 ^915 5
 rule 6 ^916 6
 rule 7 ^917 7
 rule 8 ^918 8
 rule 9 ^919 9
!
!
dial-peer voice 2 voip
 destination-pattern 91..........
 translate-outgoing called 1
 session target ras
```

The configuration above results in the stripping of the leading digits 91 from any called number that begins with 91 before the number is forwarded by the outbound VoIP dial peer. Use the caret (^) symbol to specify that the matched digits must occur at the start of a dial string.

Note Wildcard symbols such as the period (.), asterisk (*), percent sign (%), plus sign (+), and question mark (?) are not valid in translation rules. The router simply ignores these symbols when converting a number if they are used in a translation rule.

Translation rules can also be used to change the numbering type for a call. For example, some gateways may tag any number with more than 11 digits as an international number, even when the user must dial a 9 to reach an outside line. The following example shows a translation rule that converts any called number that starts with 91, and that is tagged as an international number, into a national number without the 9 before sending it to the PSTN.

```
translation-rule 20
 rule 1 91 1 international national
!
!
dial-peer voice 10 pots
 destination-pattern 91..........
 translate-outgoing called 20
 port 1:D
!
```

Note Using digit translation rules with the **num-exp** or **prefix** command is not recommended unless it is the only way to minimize confusion.

Configuring Digit Translation Rules

To create digit translation rules, perform the tasks in the following procedure:

- Creating Digit Translation Rules (Required)

To apply digit translation rules to VoIP calls, perform one or more of the following procedures:

- Applying Translation Rules to Inbound POTS Calls (Optional)
- Applying Translation Rules to Inbound VoIP Calls (Optional)
- Applying Translation Rules to Outbound Call Legs (Optional)

Creating Digit Translation Rules

To enter translation-rule configuration mode and specify a set of translation rules, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# translation-rule name-tag`	Defines a digit translation-rule set and enters translation-rule configuration mode. All subsequent commands that you enter in this mode before you exit will apply to this translation-rule set. The *name-tag* argument represents a unique number that identifies the set of translation rules. Valid entries are from 1 to 2147483647.
Step 2	`Router(config-translate)# rule name-tag input-matched-pattern substituted-pattern [match-type substituted-type]`	Defines an individual translation rule. This command can be entered up to 11 times to add an individual translation rule to the translation rule set defined in Step 1.
		The *name-tag* argument represents a unique number that identifies this individual translation rule. Valid entries are from 0 to 10. The *input-matched-pattern* argument defines the digit string that must be matched, and then replaced with the *substituted-pattern*. The *substituted-pattern* argument defines the digit string that replaces the *input-matched-pattern*. The optional *match-type* argument defines the numbering-type that you want to replace with the numbering-type defined in *substituted-type*. Enter **any** for the *match-type* if you want to match on any numbering-type.

Command	Purpose
	Otherwise, enter one of the following keywords for each of these arguments: • **abbreviated** • **international** • **national** • **network** • **reserved** • **subscriber** • **unknown** For a description of these numbering-types, see the "Numbering Type Matching" section on page 183.

To create additional individual translation rules to include in the translation-rule set, repeat Step 2.

Note Applying translation rules to more than one call leg in an end-to-end call is not recommended.

Applying Translation Rules to Inbound POTS Calls

To apply a translation rule set to all inbound POTS calls that terminate on the same voice port, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice-port` *location*	Specifies the voice port through which the call enters the router. The **voice-port** command syntax is platform-specific. For more information about the syntax of this command, see the "Voice Port Configuration" chapter.
Step 2	`Router(config-voiceport)# translate {called | calling}` *name-tag*	Specifies the translation rule set to apply to the called number or calling number. The **called** keyword applies the translation rule to the called party number. The **calling** keyword applies the translation rule to the calling party number. The *name-tag* argument is the reference number of the translation rule. Valid entries are 1 through 2147483647.

Note When this method is used, the digit translation rules are executed first before the inbound POTS dial peer is matched.

Applying Translation Rules to Inbound VoIP Calls

To apply a translation rule set to all inbound VoIP calls that originate at an H.323 gateway, use the following command in global configuration mode:

Command	Purpose
`Router(config)# voip-incoming translation-rule {called | calling} name-tag`	Specifies the translation rule set to apply to all inbound VoIP call legs that originate from an H.323 gateway. The **called** keyword applies the translation rule to the called party number. The **calling** keyword applies the translation rule to the calling party number. The *name-tag* argument is the reference number of the translation rule. Valid entries are 1 through 2147483647.

Note When using this method, the digit translation rules are executed first before the inbound VoIP dial peer is matched.

Note Digit translation rules are not supported for inbound session initiation protocol (SIP) calls.

Applying Translation Rules to Outbound Call Legs

To apply a translation rule set to an outbound VoIP or POTS call leg, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# dial-peer voice number voip`
	or
	`Router(config)# dial-peer voice number pots`
		Enters dial-peer configuration mode to configure a VoIP dial peer.
		Enters dial-peer configuration mode to configure a POTS dial peer.

Step 2	`Router(config-dial-peer)# translate-outgoing {called | calling} name-tag`	Specifies the translation rule set to apply to the calling number or called number. The **called** keyword applies the translation rule to the called party number. The **calling** keyword applies the translation rule to the calling party number. The *name-tag* argument is the reference number of the translation rule. Valid entries are 1 through 2147483647.

Note Translation rules that are configured in a dial peer using the **translate-outgoing** command are not applied to inbound call legs. When using two-stage dialing, the translation rules that are configured in the voice port using the **translate** command are applied twice; after the inbound dial peer is matched, and again after the digits are collected.

Note If the **prefix** command is also configured in the dial peer, the **translate-outgoing** command is executed first.

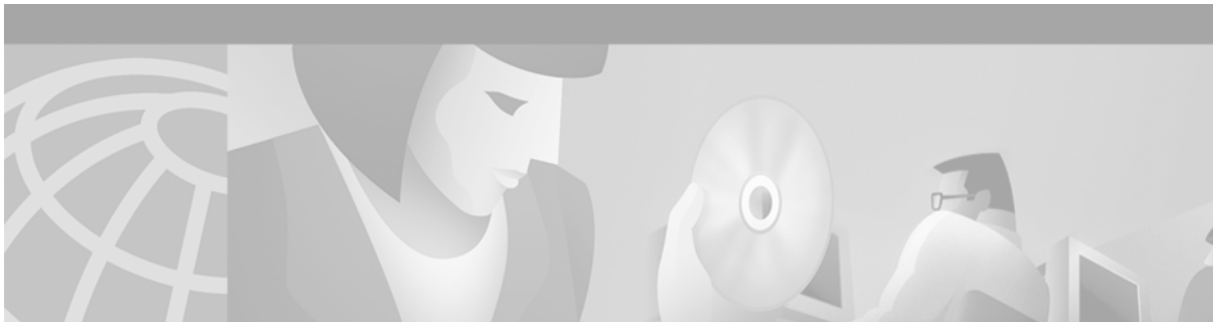
Verifying Digit Translation

To verify the configuration of a digit translation rule, enter the **show translation-rule** EXEC command. The following example shows output for a specific translation rule:

```
Router# show translation-rule 10

Translation rule address: 0x62C4F4B0
Tag name: 10
Translation rule in_used 1
**** Xrule rule table *******
        Rule : 1
        in_used state: 1
        Match pattern: 555.%
        Sub  pattern: 1408555
        Match  type: subscriber
        Sub type: international
**** Xrule rule table *******
        Rule : 2
        in_used state: 1
        Match pattern: 91.%
        Sub  pattern: 1
        Match  type: international
        Sub type: national
**** Xrule rule table *******
        Rule : 3
        in_used state: 1
        Match pattern: 527.%
        Sub  pattern: 1408527
        Match  type: subscriber
        Sub type: international
```

To verify whether a digit translation rule functions as expected, enter the **test translation-rule** EXEC command. The following example shows that when translation rule 10 is used, the number 5551212 is converted to 14085551212:

```
Router# test translation-rule 10 5551212

The replaced number: 14085551212
```



Configuring Quality of Service for Voice

This chapter describes quality of service (QoS) for voice and has the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

QoS for Voice Overview

Networks today are carrying more data than ever in the form of bandwidth-intensive, real-time voice, video, and data, which stretch network capability and resources. Cisco IOS software provides QoS solutions that help to solve problems caused by increasing traffic demands on a network.

QoS refers to the ability of a network, whether the network is a complex network, small corporate network, Internet service provider (ISP), or enterprise network, to provide better service to selected network traffic over various technologies, including Frame Relay, ATM, Ethernet and 802.1 networks, and SONET, as well as IP-routed networks that may use any or all of these underlying technologies.

The primary goals of QoS are to provide better and more predictable network service by providing dedicated bandwidth, controlled jitter and latency, and improved loss characteristics. QoS achieves these goals by providing tools for managing network congestion, shaping network traffic, using expensive wide-area links more efficiently, and setting traffic policies across the network.

QoS provides these benefits:

- Control over bandwidth, equipment, and wide-area facilities. As an example, you can limit the bandwidth consumed over a backbone link by file transfer protocol (FTP) or queueing of an important database access.
- More efficient use of network resources—Network analysis management and accounting tools, enable you to know what your network is being used for and ensure that you are servicing the most important traffic to your business.

- Tailored services—QoS enables ISPs to offer carefully tailored grades of service differentiation to their customers.
- Coexistence of mission-critical applications—Cisco QoS technologies make certain that bandwidth and minimum delays required by time-sensitive multimedia and voice applications are available; and that other applications using the link get their fair service without interfering with mission-critical traffic.
- Foundation for a fully integrated network—Cisco QoS technologies fully integrates a multimedia network, for example, by implementing weighted fair queueing (WFQ) to increase service predictability and IP Precedence signaling for traffic differentiation. Also available is ReSerVation Protocol (RSVP), which allows you to take advantage of dynamically signaled QoS.

The basic QoS architecture has three components necessary to deliver QoS across a network comprising heterogeneous technologies (IP, ATM, LAN switches, etc.) as follows:

- QoS within a single network element (for example, queueing, scheduling, and traffic shaping tools)
- QoS signaling techniques for coordinating QoS from end-to-end between network elements
- QoS policy, management, and accounting functions to control and administer end-to-end traffic across a network

The next section describes the tools that Cisco IOS software provides in each section of the architecture, which, when combined, can create end-to-end QoS or simply solve specific problems at various points in the network.

For more information regarding the concepts and complexities of QoS, refer to the *Cisco IOS Quality of Service Solutions Configuration Guide* and *Cisco IOS Quality of Service Solutions Command Reference*. For more information about the configuration of playout delay (jitter), see the "Configuring Voice Ports" chapter; and, for information about dial peers, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

QoS for Voice Tools

Cisco offers many tools for implementing QoS for voice. In general, each network has individual problems that you can solve using one or more of Cisco QoS tools. Voice over IP (VoIP) comes with its own set of problems (packet loss, jitter, and handling delay) and QoS can help solve some of these problems. Some of the problems QoS *cannot* solve are propagation delay, codec delay, sampling delay, and digitalization delay.

The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) G.114 recommendation suggests no more than 150 milliseconds (ms) of end-to-end delay to maintain good voice quality.

This section contains a high-level overview of the following:

Edge Functions

As VoIP networks are designed, edge functions usually correspond to wide-area networks (WANs) that have less than a T1 or E1 line of bandwidth from the central site. The following concepts are discussed:

Bandwidth Limitations

The first issue of major concern when designing a VoIP network is bandwidth constraints. Depending upon which codec you use and how many voice samples you want per packet, the amount of bandwidth per call can increase drastically. For a list of bandwidth consumed by codec, see Table 19.

Table 19 Codec Type and Sample Size Effects on Bandwidth

Codec	Bandwidth Consumed	Bandwidth Consumed with cRTP (2-Byte Header)	Sample Latency
G.729 with one 10-ms sample per frame	40 kbps	9.6 kbps	10 ms
G.729 with four 10-ms samples per frame	16 kbps	8.4 kbps	40 ms
G.729 with two 10-ms samples per frame	24 kbps	11.2 kbps	20 ms
G.711 with one 10-ms sample per frame	112 kbps	81.6 kbps	10 ms
G.711 with two 10-ms samples per frame	96 kbps	80.8 kbps	20 ms

In the table, 24 kbps of bandwidth is consumed when an 8-kbps codec is used. The amount of consumed bandwidth is affected by the codec used. For example, if the G.729 codec is used for two 10-ms samples, the amount of bandwidth consumed is 20 bytes per frame, which works out to 8 kbps. The packet headers that include IP, RTP, and User Datagram Protocol (UDP) add 40 bytes to each frame. The header is *twice* the amount of the payload.

If the G.729 codec is used with two 10-ms samples, without RTP header compression, 24 kbps are consumed in each direction per call. Although this might not be a large amount for T1 (1.544-mbps), E1 (2.048-mbps), or higher circuits, it is a large amount (42 percent) for a 56-kbps circuit.

Also, the bandwidth does not include layer 2 headers (PPP, Frame Relay, etc.). It includes headers from layer 3 (network layer) and above only. Therefore, the same G.729 call can consume different amounts of bandwidth based upon which data link layer is used (Ethernet, Frame Relay, PPP, and etc.).

Real-Time Transport Protocol

To reduce the large percentage of bandwidth consumed by a G.729 voice call, you can use compressed Real-Time Transport Protocol (cRTP). cRTP enables you to compress the 40-byte IP/RTP/UDP header to 2 to 4 bytes most of the time.

With cRTP, the amount of traffic per VoIP call is reduced from 24 kbps to 11.2 kbps. This is a major improvement for low-bandwidth links. A 56-kbps link, for example, can now carry four G.729 VoIP calls at 11.2 kbps each. Without cRTP, only two G.729 VoIP calls at 24 kbps can be used.

To avoid the unnecessary consumption of available bandwidth, cRTP is used on a link-by-link basis. This compression scheme reduces the IP/RTP/UDP header to 2 bytes when UDP checksums are not used, or 4 bytes when UDP checksums are used.

cRTP uses some of the same techniques as TCP header compression. In TCP header compression, the first factor-of-two reduction in data rate occurs because half of the bytes in the IP and TCP headers remain constant over the life of the connection.

The big gain, however, comes from the fact that the difference from packet to packet is often constant, even though several fields change in every packet. Therefore, the algorithm can simply add 1 to every value received. By maintaining both the uncompressed header and the first-order differences in the session state shared between the compressor and the decompressor, cRTP must communicate only an indication that the second-order difference is zero. In that case, the decompressor can reconstruct the original header without any loss of information, simply by adding the first-order differences to the saved, uncompressed header as each compressed packet is received.

Just as TCP/IP header compression maintains shared state for multiple simultaneous TCP connections, this IP/RTP/UDP compression must maintain state for multiple session contexts. A *session context* is defined by the combination of the IP source and destination addresses, the UDP source and destination ports, and the RTP synchronization source (SSRC) field. A compressor implementation might use a hash function on these fields to index a table of stored session contexts.

The compressed packet carries a small integer, called the *session context identifier*, or CID, to indicate in which session context that packet should be interpreted. The decompressor can use the CID to index its table of stored session contexts.

cRTP can compress the 40 bytes of header down to 2 to 4 bytes most of the time. As such, about 98 percent of the time the compressed packet will be sent. Periodically, however, an entire uncompressed header must be sent to verify that both sides have the correct state. Sometimes, changes occur in a field that is usually constant, such as the payload type field. In such cases, the IP/RTP/UDP header cannot be compressed, so an uncompressed header must be sent.

You should use cRTP on any WAN interface where voice bandwidth is a concern and a high proportion of RTP traffic exists.

Queueing

Queueing is like the concept of first in first out (FIFO), which means that the first in line is the first to get out of the line. FIFO queueing was the first type of queueing to be used in routers, and it is still useful, depending upon the network topology. In networks today, with a variety of applications, protocols, and users, a way to classify different traffic is required.

Cisco has several queueing tools that enable a network administrator to specify what type of traffic is special or important and to queue the traffic based upon that information. The most popular technique is WFQ.

There are the several queueing types that are listed below. For more information, see the *Cisco IOS Quality of Service Solutions Configuration Guide* and *Cisco IOS Quality of Service Solutions Command Reference*:

- Weighted fair queueing
- Custom queueing
- Priority queueing
- Class-based (CB) WFQ
- Priority queuing (PQ) with CB-WFQ

Packet Classification

To achieve your intended packet delivery, you must know how to properly weight WFQ. There are different weighting techniques and ways to use them in various networks to achieve the degree of QoS you require.

IP Precedence

IP Precedence is a value defined by the three bits in the type of service (ToS) field in an IP header. IP Precedence enables a router to group traffic flows based upon the eight precedence settings and to queue traffic based upon that information as well as on source address, destination address, and port numbers.

Policy Routing

Policy routing is a routing scheme that forwards packets to specific interfaces based on user-configured policies. Such policies might specify that traffic sent from a particular network should be forwarded out one interface, while all other traffic should be forwarded out another interface.

With policy-based routing, you can configure a defined policy for traffic flows and not have to rely completely on routing protocols to determine traffic forwarding and routing. Policy routing also enables you to set the IP Precedence field so that the network can utilize different classes of service.

You can base policies on IP addresses, port numbers, protocols, or the size of packets. You can use one of these descriptors to create a simple policy, or you can use all of them to create a complicated policy.

All packets received on an interface with policy-based routing enabled are passed through enhanced packet filters known as *route maps*. The route maps dictate where the packets are forwarded.

RSVP

RSVP enables endpoints to signal the network with the kind of QoS needed for a particular application. Most networks are designed to assume what QoS applications require. Network administrators can use RSVP as *dynamic access lists*. Using RSVP means that network administrators do not need to be concerned with port numbers of IP packet flows because RSVP signals that information during its original request.

RSVP is an out-of-band, end-to-end signaling protocol that requests a certain amount of bandwidth and latency with each network hop that supports RSVP. If a network node (router) does not support RSVP, RSVP moves onto the next hop. A network node has the option to approve or deny the reservation based upon the load of the interface to which the service is requested.

VoIP Call Admission Control

Cisco VoIP call admission control (CAC) applications use RSVP to limit the accepted voice load on the IP network and guarantee the QoS levels of calls. The VoIP CAC using RSVP synchronizes RSVP signaling with Cisco H.323 Version 2 signaling to ensure that the bandwidth reservation is established in both directions before a call moves to the alerting phase (ringing). This ensures that the called party phone rings only after the resources for the call have been reserved. Using RSVP-based admission control, VoIP applications can reserve network bandwidth and react appropriately if bandwidth reservation fails.

Prior to Cisco IOS release 12.1(3)XI and 12.1(5)T, VoIP gateways used H.323 Version 1 (Slow Connect) procedures when initiating calls requiring bandwidth reservation. This feature, which is enabled by default, allows gateways to use H.323 Version 2 (Fast Connect) for all calls, including those requiring RSVP.

To enable backward compatibility, commands are available to force the originating gateway to initiate calls using Slow Connect procedures if the terminating gateway is running Cisco IOS Release 12.1(1)T or later. You can configure Slow Connect globally for all VoIP calls by using the **h323 call start** voice-service configuration command, or configure Slow Connect per individual VoIP dial peer by using the **call start** voice-class configuration command.

A timer can be set by using the **call rsvp-sync serv-timer** command to limit the number of seconds that the terminating gateway waits for bandwidth reservation setup before proceeding with the call setup or releasing the call, depending on the configured QoS level in the dial peers.

Synchronized RSVP is attempted for a VoIP call as long as the requested (desired) QoS for the associated dial peer is set to controlled-load or guaranteed-delay. If the requested QoS level is set to the default of best-effort, bandwidth reservation is not attempted. If RSVP reservation is attempted but fails, the acceptable QoS for the dial peer determines the outcome of the call. When the acceptable QoS is configured for best effort, the call setup proceeds, but without any bandwidth reservation in place. When the acceptable QoS on either gateway is configured for other than best effort, and the RSVP reservation fails, the call is released. The requested QoS and acceptable QoS are configured through Cisco IOS software by using the **req-qos** and **acc-qos** dial-peer configuration commands, respectively.

Table 20 summarizes the results of nine call setup scenarios using Fast Connect, based on the QoS levels configured in the VoIP dial peers at the originating and terminating gateways. The table does not include cases in which the requested QoS is best-effort and the acceptable QoS is other than best-effort and is valid only for calls using Fast Connect procedures.

Table 20 Call Results Based on Configured QoS Levels

Call Scenarios	Originating Gateway		Terminating Gateway		
	Requested QoS	**Acceptable QoS**	**Requested QoS**	**Acceptable QoS**	**Results**
1	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	Call proceeds only if both RSVP reservations succeed.
2	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	best-effort	Call proceeds only if both RSVP reservations succeed.
3	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	best-effort	best-effort	Call is released.
4	controlled-load or guaranteed-delay	best-effort	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	Call proceeds only if both RSVP reservations succeed.
5	controlled-load or guaranteed-delay	best-effort	controlled-load or guaranteed-delay	best-effort	Call proceeds regardless of RSVP results. If RSVP reservation fails, call receives best-effort service.
6	controlled-load or guaranteed-delay	best-effort	best-effort	best-effort	Call proceeds with best-effort service.

Table 20 Call Results Based on Configured QoS Levels (continued)

Call Scenarios	Originating Gateway		Terminating Gateway		
	Requested QoS	Acceptable QoS	Requested QoS	Acceptable QoS	Results
7	best-effort	best-effort	controlled-load or guaranteed-delay	controlled-load or guaranteed-delay	Call is released.
8	best-effort	best-effort	controlled-load or guaranteed-delay	best-effort	Call proceeds with best-effort service.
9	best-effort	best-effort	best-effort	best-effort	Call proceeds with best-effort service.

The following are the benefits of using CAC with RSVP:

- VoIP gateways default to H.323 Version 2 (Fast Connect) for all calls.
- Called-party phone rings only after bandwidth reservation is confirmed.
- QoS for voice calls is guaranteed across the IP network.

The following are restrictions on VoIP CAC using RSVP:

- To support RSVP-based QoS with H.323 Version 2 (Fast Connect), the originating and terminating gateways must be running Cisco IOS Release 12.1(3)XI or 12.1(5)T, or later.
- To support RSVP-based QoS with H.323 Version 1 (Slow Connect), Cisco H.323 Version 2 gateways must be running Cisco IOS Release 12.1(1)T or later.
- RSVP with multicast is not supported.

IP RTP Priority

When WFQ is enabled and IP RTP Priority is configured, a strict priority queue is created. You can use IP RTP Priority to enable use of the strict priority queueing scheme for delay-sensitive data. You can identify voice traffic by its UDP port numbers and classify it into a priority queue. The result is voice traffic that has strict priority service in preference to all other traffic. This is the most highly recommended classification scheme for VoIP networks on lower-bandwidth links (768 kbps and below).

Traffic Policing for Voice Networks

The preceding sections cover ways you can queue different flows of traffic and then prioritize those flows. That is an important part of QoS. Sometimes, however, it is necessary to actually regulate or limit the amount of traffic an application is allowed to send across various interfaces or networks.

Cisco IOS software has a few tools that enable network administrators to define how much bandwidth an application or even a user can use. These features have two different tools: *rate-limiting* and *shaping*.

The main difference between these two traffic-regulation tools is that rate-limiting tools drop traffic based upon policing, and shaping tools generally buffer the excess traffic while waiting for the next open interval to transmit the data.

The similarities are in that both the rate-limiting and shaping tools identify when traffic exceeds the thresholds set by the network administrator. Often, these two tools are used together. Traffic shaping is used at the edge of the network (on customer premises) to make sure the customer is utilizing the bandwidth for business needs. Rate-limiting tools often used in service provider networks to ensure that a subscriber does not exceed the amount of bandwidth set by contract with the service provider.

You can rate-limit traffic by precedence, Media Access Control (MAC) address, IP addresses, or other parameters. Network administrators also can configure access lists to create even more granular rate-limiting policies.

Traffic Shaping for Voice Networks

Cisco IOS QoS software includes two types of traffic-shaping tools: Generic Traffic Shaping (GTS) and Frame Relay traffic shaping (FRTS). The two traffic-shaping methods are similar in implementation, although their command-line interfaces differ somewhat and they use different types of queues to contain and shape traffic that is deferred.

If a packet is deferred, GTS uses a WFQ to hold the delayed traffic. FRTS uses either a custom queue (CQ) or a priority queuing (PQ) to hold the delayed traffic, depending on what you configured. FRTS also supports WFQ to hold delayed traffic.

Traffic shaping enables you to control the traffic going out of an interface to match its flow to the speed of the remote, target interface and to ensure that the traffic conforms to policies contracted for it. Thus, you can shape traffic adhering to a particular profile to meet downstream requirements, thereby eliminating bottlenecks in topologies with data-rate mismatches.

You use traffic shaping primarily to do the following:

- Control usage of available bandwidth
- Establish traffic policies
- Regulate traffic flow to avoid congestion

You can also use traffic shaping to do the following:

- Configure an interface if you have a network with different access rates. Suppose one end of the link in a Frame Relay network runs at 256 kbps and the other end runs at 128 kbps. Sending packets at 256 kbps could cause the applications using the link to fail.
- Configure an interface to offer a subrate service. In this case, traffic shaping enables you to use the router to partition your T1 or T3 links into smaller channels.

Traffic shaping prevents packet loss. It is especially important to use traffic shaping in Frame Relay networks because the switch cannot determine which packets take precedence and, therefore, which packets should be dropped when congestion occurs. It is critical for VoIP that you control latency. By limiting the amount of traffic and traffic loss in the network, you can smooth out traffic patterns and give priority to real-time traffic.

High-Speed Transport

High-speed transport is defined as any interface higher than T1 speed. The QoS mechanisms required to configure a high-speed transport are as follows:

- Packet Over SONET/SDH (POS)—Prioritizes traffic on this high-speed interface up to OC-48.
- Modified deficit round robin (MDRR)—Extends Deficit Round Robin (DRR) to provide priority for real-time traffic such as VoIP. Within MDRR, IP packets are mapped to different CoS queues based on precedence bits. All the queues are serviced in round-robin fashion except for one: the priority queue used to handle voice traffic.
- IP and ATM—Maps IP prioritization onto ATM by configuring precedence values to an IP packet to different ATM PVCs. The IP prioritization enables the network administrator to have different PVCs, allocating more important traffic over a variable bit rate (VBR) ATM circuit and less important traffic over an unspecified bit rate (UBR) ATM circuit; or IP prioritization onto ATM using queueing techniques such as WFQ to prioritize different flows by PVC.

Refer to the *Cisco IOS Quality of Service Configuration Guide* and *Cisco IOS Quality of Service Command Reference* for detailed information.

Congestion Avoidance

Congestion avoidance works by dropping packets from different flows, causing applications to slow the amount of traffic being sent.

WRED

Random Early Detection (RED) is a congestion avoidance mechanism, and Weighted RED (WRED) is the Cisco IOS software implementation of dropping traffic to avoid global synchronization. WRED combines the capabilities of the RED algorithm with IP precedence. This combination provides for preferential traffic handling for higher-priority packets. It can selectively discard lower-priority traffic when the interface starts to get congested and provide differentiated performance characteristics for different classes of service. To fully comprehend how WRED works, you must understand TCP packet-loss behavior.

TCP

A stream of data sent on a TCP connection is delivered reliably and in order to the destination. Transmission is made reliable through the use of sequence numbers and acknowledgments. Segments (segments sequentially numbered) carry an acknowledgment number, which is the sequence number of the next expected data octet of transmissions in the reverse direction. When the TCP transmits a segment, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer runs out, the segment is retransmitted.

To govern the flow of data into a TCP, flow control mechanisms are used. The data-receiving TCP reports a window to the sending TCP. This window specifies the number of octets, starting with the acknowledgment number that the data-receiving TCP is currently prepared to receive.

QoS for Voice Configuration Prerequisites

The following are tasks that must be performed prior to configuring QoS for voice:

- Establish a working IP network. For information about configuring IP, see the *Cisco IOS IP Routing Configuration Guide*.
- Configure your VoIP gateway for H.323. To support RSVP-based QoS with H.323 Version 2 (Fast Connect), the originating and terminating gateways must be running Cisco IOS Release 12.1(3)XI or 12.1(5)T, or later. For information about configuring the gateway, refer to the *Software Configuration Guide for Cisco 3600 and Cisco 2600 Series Routers* or *Configuring H.323 VoIP Gateway for Cisco Access Platforms* and the "Configuring H.323 Gateways" chapter.
- Enable RSVP on the appropriate interfaces on your gateways by using the **ip rsvp bandwidth** interface configuration command. You must also enable fair queueing on these interfaces by using the **fair-queue** interface configuration command. For information about enabling RSVP and fair queueing, refer to the *Cisco IOS Quality of Service Solutions Command Reference*.
- Set the QoS levels in your dial peers by using the **req-qos** and **acc-qos** dial-peer configuration commands. For information about configuring QoS levels, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

Note An inbound plain old telephone service (POTS) dial peer is not required if the originating and terminating gateways have outbound VoIP dial peers configured to reach the calling number at the far end and if the VoIP dial peers use the same QoS parameters. Configure an inbound POTS dial peer if the corresponding outbound VoIP dial peers at the originating and terminating gateways do not have matching QoS configurations, or if calls can be established in only one direction (for example, or if calls can be made from gateway A to gateway B, but not from gateway B to gateway A).

For information on how to configure playout delay, echo cancellation, and voice levels, see the "Configuring Voice Ports" chapter.

QoS for Voice Configuration Task List

Refer to the *Cisco IOS Quality of Service Configuration Guide* and *Cisco IOS Quality of Service Command Reference* for tasks that enable QoS for your network. To configure the H.323 Gateway, see the "Configuring H.323 Gateways" chapter.

The following sections describe optional configuration tasks for the VoIP Call Admission Control Using RSVP feature. The tasks in the first section are for configuring synchronization:

- Configuring Synchronization and the Reservation Timer, page 209 (Optional)

Use the following tasks only if you require backward compatibility with H.323, Version 2 (Slow Connect) gateways running a release earlier than Cisco IOS Release 12.1(3)XI or 12.1(5)T (must be Cisco IOS Release 12.1(1)T or later):

- Configuring Slow Connect for VoIP Globally, page 209 (Optional)
- Configuring Slow Connect for a Specific Dial Peer, page 210 (Optional)

Configuring Synchronization and the Reservation Timer

Synchronization between RSVP and the H.323 voice signaling protocol is enabled by default; no configuration tasks are required to enable this feature. To enable the feature if the **no call rsvp sync** command was used to disable it, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call rsvp sync`	Enables synchronization between RSVP and the H.323 voice signaling protocol.
Step 2	`Router(config)# call rsvp-sync resv-timer` *seconds*	Sets the timer for reservation requests. The default is 10 seconds.

Configuring Slow Connect for VoIP Globally

To make an H.323 gateway backward-compatible with a destination gateway, use the following commands beginning in global configuration mode. This procedure is optional and selects Slow Connect globally for all VoIP services.

	Command	Purpose
Step 1	`Router(config)# voice service voip`	Enters voice-service configuration mode for VoIP services.
Step 2	`Router(conf-voi-serv)# h323 call start slow`	Forces the H.323 gateway to use Slow Connect procedures. **Note** To restore the default of Fast Connect, use the **h323 call start fast** command.

Note The previous procedure selects Slow Connect globally for all VoIP calls. To change the type of connect procedures for calls associated with a specific dial peer, use the following procedure:

Configuring Slow Connect for a Specific Dial Peer

Note This procedure is optional and selects Slow Connect for a specific VoIP dial peer.

To make an H.323 gateway backward-compatible with a destination gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class h323 tag`	Enters voice-class configuration mode and creates a voice class for H.323 attributes.

Step 2	`Router(config-class)# call start slow`
	or
	`Router(config-class)# call start system`
		Forces the H.323 gateway to use Slow Connect procedures. The default is **slow**. The keyword **system** causes the H.323 gateway to use the connect procedure that is configured in the voice-service configuration (see Configuring Slow Connect for VoIP Globally).
		Note If you require Fast Connect for a specific dial peer, use the **call start fast** command to restore the default when configuring the Slow Connect for VoIP globally.

Step 3	`Router(config-class)# exit`	Exits voice-class configuration mode and returns to global configuration mode.
Step 4	`Router(config)# dial-peer voice number voip`	Enters dial-peer configuration mode for the VoIP dial peer.
Step 5	`Router(config-dial-peer)# voice-class h323 tag`	Assigns the voice class attributes to the dial peer, including the H.323 connect procedure that was selected in Step 2.

Verifying the RSVP CAC Configuration

To verify that RSVP-based call admission control is configured correctly, enter the **show running-config** privileged EXEC command to display the command settings for the router, as shown in the "QoS for Voice Configuration Examples" section.

Monitoring and Maintaining RSVP Call Admission Control

To display the configuration parameters for RSVP synchronization and statistics for calls that initiate RSVP, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show call rsvp-sync conf`	Displays the RSVP synchronization configuration.
Step 2	`Router# show call rsvp-sync stats`	Displays statistics for calls that attempted RSVP reservation.

QoS for Voice Configuration Examples

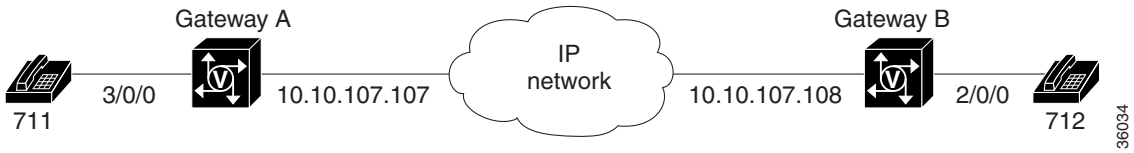
Refer to the *Cisco IOS Quality of Service Configuration Guide* and *Cisco IOS Quality of Service Command Reference* for more information.

The following examples display the screen output using the **show running-config** command:

- RSVP Synchronization Examples, page 211
- H.323 Slow Connect by Voice Service Example, page 212
- H.323 Slow Connect by Dial Peer Example, page 212

RSVP Synchronization Examples

The following examples show that calls can be made in either direction between gateway A and gateway B, which are connected to POTS phones, with phone numbers 711 and 712, respectively. The requested QoS indicates that RSVP setup must be complete before the destination phone rings. The acceptable QoS indicates that the call is released if the RSVP setup fails or is not complete within the allotted time.



Gateway A	Gateway B
call rsvp-sync call rsvp-sync resv-timer 15 ! interface Ethernet0/0 ip address 10.10.107.107 10.255.255.255 fair-queue 64 256 31 ip rsvp bandwidth 1000 1000 ! voice-port 3/0/0 ! dial-peer voice 712 voip destination-pattern 712 session target ipv4:10.10.107.108 req-qos controlled-load acc-qos controlled-load ! dial-peer voice 711 pots destination-pattern 711 port 3/0/0	call rsvp-sync call rsvp-sync resv-timer 15 ! interface Ethernet0/0 ip address 10.10.107.108 10.255.255.255 fair-queue 64 256 31 ip rsvp bandwidth 1000 1000 ! voice-port 2/0/0 ! dial-peer voice 711 voip destination-pattern 711 session target ipv4:10.10.107.107 req-qos controlled-load acc-qos controlled-load ! dial-peer voice 712 pots destination-pattern 712 port 2/0/0

H.323 Slow Connect by Voice Service Example

The following example shows that Slow Connect is configured globally for all VoIP calls because the **h323 call start slow** command is used in the voice service configuration:

```
dial-peer voice 712 voip
 destination-pattern 712
 session target ipv4:10.10.107.108
 req-qos controlled-load
 acc-qos controlled-load
!
voice service voip
 h323 call start slow
```

The following example shows the same basic configuration but demonstrates that when the **call start system** command is used in the voice class configuration, the gateway defaults to the connect procedure that is configured in the voice service; otherwise the dial peer configuration takes precedence (see the section "H.323 Slow Connect by Dial Peer Example").

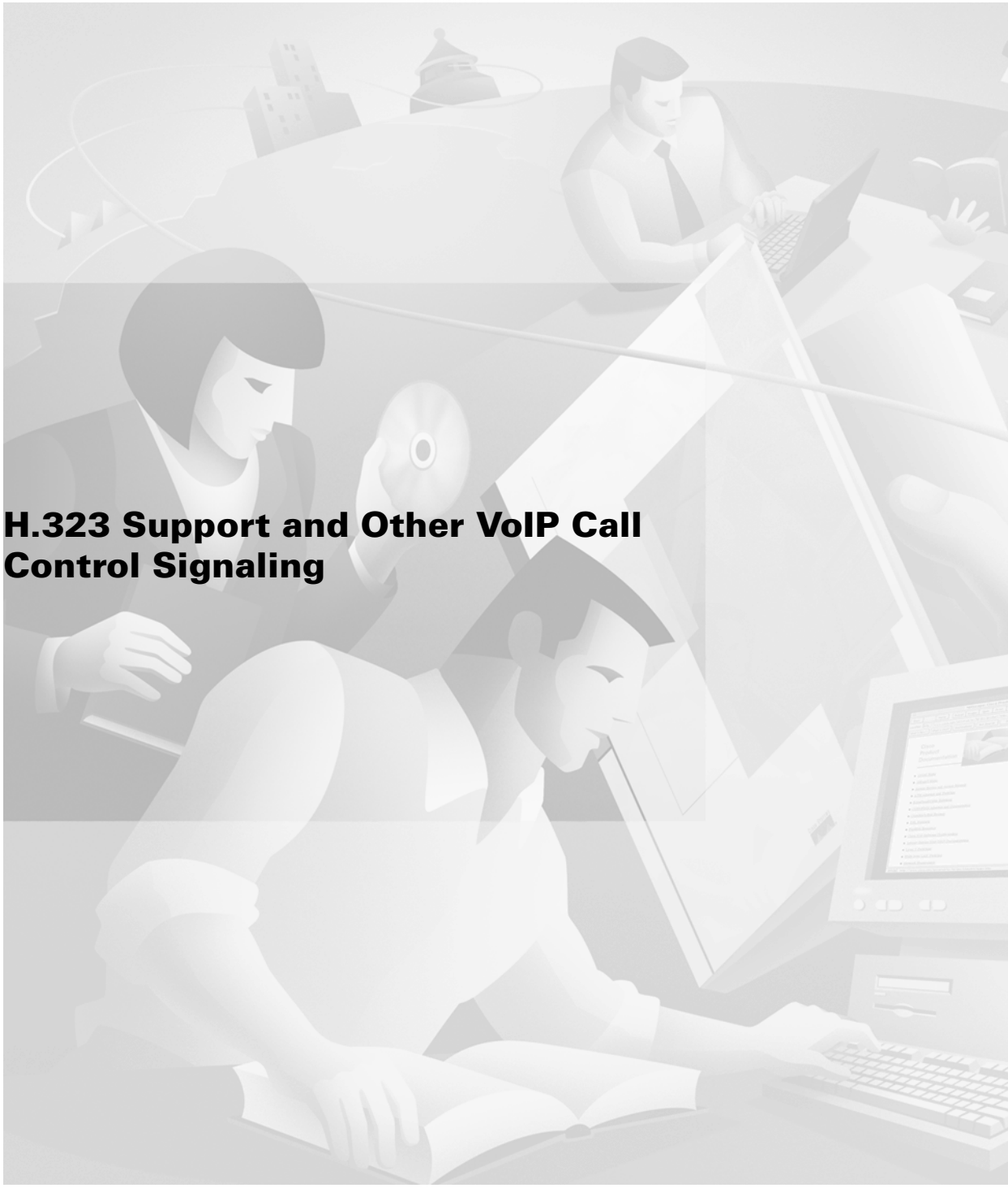
```
dial-peer voice 712 voip
 voice-class h323 2
 destination-pattern 712
 session target ipv4:10.10.107.108
 req-qos controlled-load
 acc-qos controlled-load
!
voice class h323 2
 call start system
!
voice service voip
 h323 call start slow
!
```

H.323 Slow Connect by Dial Peer Example

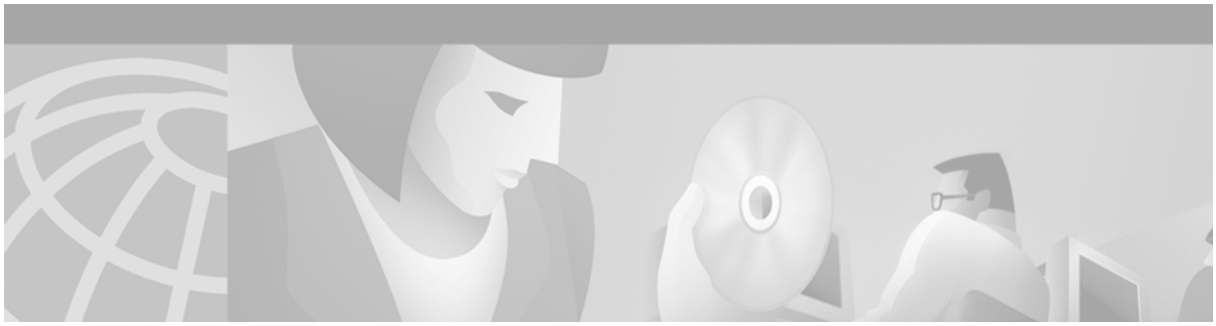
The following example shows that calls from VoIP dial peer 712 use Slow Connect procedures because the **call start slow** command is configured in the voice class assigned to the dial peer:

```
dial-peer voice 712 voip
 voice-class h323 2
 destination-pattern 712
 session target ipv4:10.10.107.108
 req-qos controlled-load
 acc-qos controlled-load
!
voice class h323 2
 call start slow
!
voice service voip
 h323 call start fast
!
```

Note The **h323 call start fast** voice-service command is ignored because the voice class configuration takes precedence, unless the **call start system** voice-class command is used (see the section "H.323 Slow Connect by Voice Service Example").



H.323 Support and Other VoIP Call Control Signaling



Configuring Media Gateway Control Protocol and Related Protocols

This chapter describes the concepts and configuration procedures for Media Gateway Control Protocol (MGCP). MGCP defines the call control relationship between Voice over IP (VoIP) gateways that translate audio signals to and from the packet network, and call agents (CAs). The CAs are responsible for processing the calls.

Note An earlier implementation of the protocol, Simple Gateway Control Protocol (SGCP), is no longer available as a standalone product. MGCP supports SGCP functionality for those customers who want SGCP capabilities. For more information on SGCP, see *Simple Gateway Control Protocol Support on the Cisco MC3810 and Cisco 3600 Series Routers*.

This chapter includes the following sections:

Cisco IOS Release 12.2 supports the MGCP 0.1, SGCP 1.1+, SIP, and H.323 protocols on these platforms:

- Cisco 2600 series modular access routers
- Cisco 3640 and 3660 multiservice platforms
- Cisco AS5300 universal access server
- Cisco uBR924 cable access router
- Cisco Voice Gateway 200 (VG200)

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

MGCP Configuration Overview

MGCP is an extension of the earlier version of the protocol SGCP and supports SGCP functionality in addition to several enhancements. Systems using SGCP can easily migrate to MGCP, and MGCP commands are available to enable the SGCP capabilities.

An MGCP gateway handles the translation between audio signals and the packet network. The gateways interact with a CA, also called a Media Gateway Controller (MGC) that performs signal and call processing on gateway calls. In the MGCP configurations that Cisco IOS supports, the gateway can be a Cisco router, access server, or cable modem, and the CA is a server from a third-party vendor.

Configuration commands for MGCP define the path between the call agent and the gateway, the type of gateway, and the type of calls handled by the gateway.

MGCP uses endpoints and connections to construct a call. Endpoints are sources of or destinations for data, and can be physical or logical locations in a device. Connections can be point-to-point or multipoint.

Similar to SGCP, MGCP uses UDP for establishing audio connections over IP networks. However, MGCP also uses *hairpinning* to return a call to the PSTN when the packet network is not available.

Creating a call connection involves a series of signals and events that describe the connection process. This information might include such indicators as the off-hook status, a ringing signal, or a signal to play an announcement. These events and signals are specific to the type of endpoint involved in the call.

MGCP groups these events and signals into packages. A trunk package, for example, is a group of events and signals relevant to a trunking gateway, while an announcement package groups events and signals for an announcement server. MGCP supports seven package types that are as follows:

- Trunk
- Line
- Dual tone multi-frequency (DTMF)
- Generic media
- Real-time Transport Protocol (RTP)
- Announcement server
- Script

The trunk package and line package are supported by default on certain types of gateways. Although configuring a gateway with additional endpoint package information is optional, you may want to specify the packages for your endpoints to add to or to override the defaults.

MGCP provides the following benefits:

- Alternative dial tone for voice over IP environments—Deregulation in the telecommunications industry gives competitive local exchange carriers (CLECs) opportunities to provide toll bypass from the incumbent local exchange carriers (ILECs) by using VoIP. MGCP enables a VoIP system to control call setup and teardown and CLASS features for less sophisticated gateways.
- Configuration requirements for static VoIP network dial peers has been removed—When MGCP is used as the call agent in a VoIP environment, configuring static VoIP network dial peers is not required, and so the configuration is simplified. The MGCP call agent provides functions similar to VoIP network dial peers.

Note POTS dial peer configuration is still required.

- Migration paths—Systems using earlier versions of the protocol can migrate easily to MGCP.
- Multiple protocols support and investment protection—Cisco IOS Release 12.2 supports concurrently on the same hardware and software the MGCP Version 0.1, SGCP 1.1+, SIP, and H.323 protocols. VoIP solutions can use any of these popular protocols. Changing protocols for new network solutions can be done without disrupting the current network or investing in new systems.
- Varied network needs supported as follows:
  - IXCs that have no legacy TDM equipment in their networks and want to deploy a fully featured network that offers both long-distance services to corporate customers and connectivity to local exchange carriers or other IXCs with traditional TDM equipment.
  - IXCs who have TDM equipment in their networks and want to relieve the congestion in the network using data technologies to carry voice traffic or to cap the growth of TDM ports. In these situations, the packet network provides basic switched trunking without services or features.
  - Competitive CLECs who want to provide residential and enhanced services.
  - Dial access customers who want enhanced SS7 access capabilities and increased performance, reliability, scalability, and lower costs.

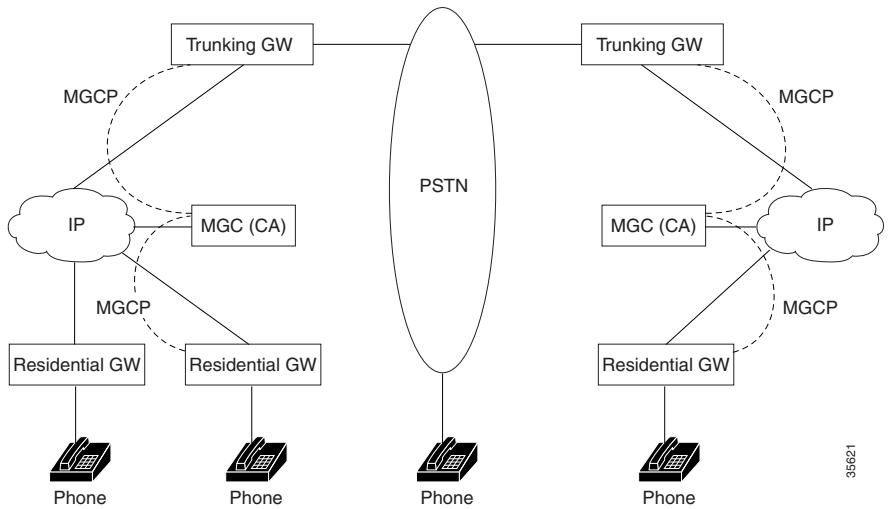
Supported Gateways

MGCP supports residential and trunking gateways and each gateway is described in the following sections.

Residential Gateway

A residential gateway (RGW) provides an interface between analog (RJ-11) calls from a telephone and the VoIP network. Examples of RGWs include cable modems and the Cisco 2600 series routers. See Figure 54 for an illustration of an RGW configuration.

Figure 54 Residential and Trunking Gateways



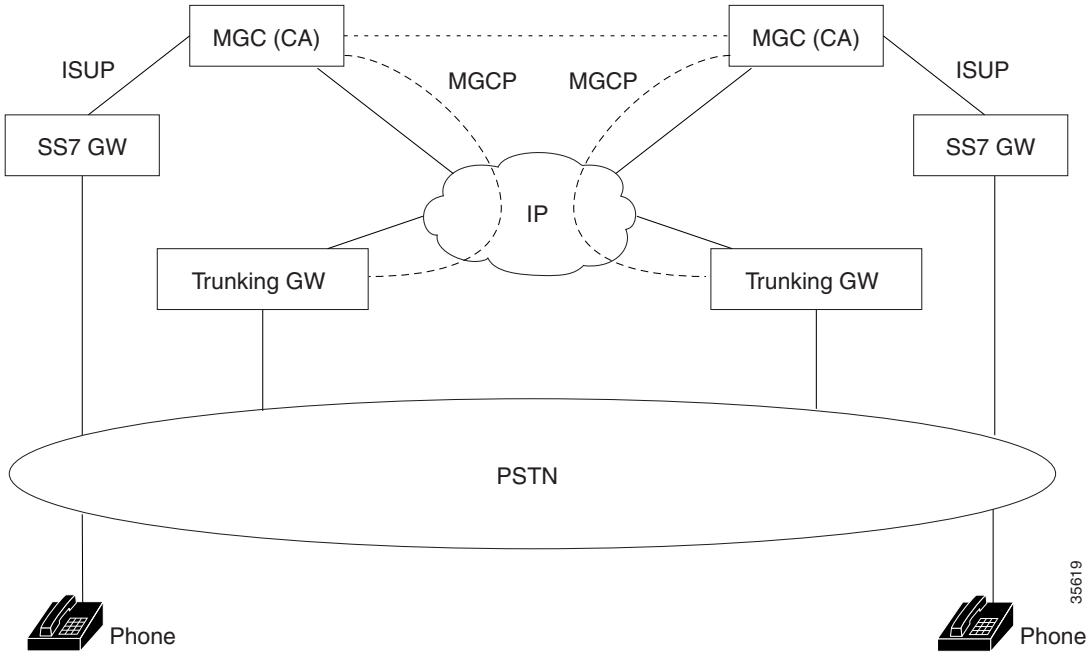
RGW functionality supports analog plain old telephone service (POTS) calls for both SGCP and MGCP on the Cisco 2600 series routers and Cisco uBR924 cable access router:

- Call waiting and stutter dial tone are supported on the Cisco 2600 series router and Cisco uBR924 cable access router.
- On-hook caller ID, distinctive ringing, and ring splash are supported only on the Cisco uBR924 cable access router.
- A default call agent address can be specified for each FXS port on the Cisco uBR924 cable access router.
- Modem and fax calls are supported on the Cisco 2600 series router and Cisco uBR924 cable access router.

Trunking Gateway

A trunking gateway (TGW) provides an interface between trunks on the public switched telephone network (PSTN) and a VoIP network. A trunk can be a DS0, T1, or E1 line. Examples of TGWs include access servers and routers. See Figure 55 for an illustration of a TGW configuration.

Figure 55 Trunking Gateways



TGW functionality supports SGCP and MGCP on the following platforms:

- Cisco AS5300 universal access servers and the Cisco 3660 router for SS7 calls.
- Cisco AS5300 universal access servers with SGCP 1.1+ protocol for Feature Group D Operator Services (FGD-OS) 911 outgoing calls on T1 lines.
- Cisco AS5300 universal access servers for PRI/ISDN signaling. These calls are backhauled to the CA.
- Cisco AS5300 universal access servers and Cisco 3660 routers for T1 and E1 interfaces.

- Cisco AS5300 universal access servers and Cisco 3660 routers for modem and fax calls.

MGCP Prerequisite Tasks

Complete the following tasks on your network before configuring MGCP:

- Configure IP routing.
- Configure voice ports.
- Configure VoIP.
- Configure the call agent.

MGCP Configuration Task List

To configure MGCP, perform the tasks in the following sections. Each task in the list is identified as either optional or required.

- Do at least one of the following tasks, depending on your network configuration (required):
  - Configuring a TGW for MGCP, page 220
  - Configuring a TGW for SGCP, page 222
  - Configuring an RGW, page 223
  - Verifying the TGW or RGW Configuration, page 225
- Blocking New Calls and Gracefully Terminating Existing Calls, page 225 (optional)
- Monitoring and Maintaining MGCP, page 225 (optional)

Note RGWs are configured only with MGCP.

Configuring a TGW for MGCP

To configure a TGW for MGCP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mgcp`	Initiates the MGCP application.
Step 2	`Router(config)# mgcp call-agent [ipaddr|hostname] [port] service-type mgcp`	Specifies the call agent's IP address or domain name, the port, and gateway control service type. The keywords and arguments are as follows: • *ipaddr*—Specifies the call agent's IP address. • *hostname*—Specifies the call agent's hostname, using the format host.domain.ext. • *port*—Specifies the port for the call agent to use. Valid values are from 1025 through 65535. • **service-type**—Specifies the type of gateway control service supported by the call agent. Valid values are **mgcp** and **sgcp**. For MGCP configurations, use **mgcp**.
Step 3	`Router(config)# controller t1 number`	Specifies the channel number of the T1 trunk to be used for analog calls.
Step 4	`Router(config-controller)# ds0-group channel-number timeslots range type none service mgcp`	Configures the channelized T1 time slots to accept the analog calls. The keywords and arguments are as follows: • *channel-number*—Specifies the DS0 group number. Valid values are from 0 to 23 for T1 interfaces and from 0 to 30 for E1 interfaces. • **timeslots** *range*—Specifies the DS0 time slot range of values. Valid values are from 1 to 24 for T1 interfaces and from 1 to 31 for E1 interfaces. The default value is 24. • **type**—Specifies the signaling type to be applied to the selected group. For MGCP functionality, use **none**. • **service**—Specifies the type of service supported on the gateway. Valid values are **mgcp** and **sgcp**. For MGCP configurations, use **mgcp**.
Step 5	`Router(config-controller)# exit`	Exits controller configuration mode and returns to global configuration mode.
Step 6	`Router(config)# mgcp restart-delay value`	(Optional) Specifies the delay value sent in the RSIP graceful teardown method. The *value* range is from 0 to 600 seconds; the default is 0.

	Command	Purpose
Step 7	`Router(config)# mgcp package-capability {s-package | dtmf-package | gm-package | rtp-package | trunk-package | script-package}`	(Optional) Specifies the event packages that are supported on the gateway. The set of supported packages varies with the type of gateway (TGW or RGW). The keywords are as follows: • **as-package**—Specifies the announcement server package. • **dtmf-package**—Specifies the DTMF package. • **gm-package**—Specifies the generic media package. • **rtp-package**—Specifies the RTP package. • **trunk-package** (default)—Specifies the trunk package. • **script-package**—Specifies the script package. Available only on the Cisco AS5300 universal access server.
Step 8	`Router(config)# mgcp default-package {as-package | dtmf-package | gm-package | rtp-package | trunk-package}`	(Optional) Specifies the event package that should act as the default. Overrides the **mgcp package-capability** default package.
Step 9	`Router(config)# mgcp dtmf-relay {codec | low-bit-rate} mode {cisco | out-of-band}`	(Optional) Used for relaying digits through the IP network. The keywords and arguments are as follows: • **codec**—Specifies use of either a G.711 or a G.726 codec. • **low-bit-rate**—Specifies a low-bit-rate codec other than G.711 and G.726. • **cisco**—Removes DTMF tone from the voice stream and sends FRF.11 with special payload 121 for DTMF digits. • **out-of-band**—Removes DTMF tone from the voice stream and does not send FRF.11. The default is **no mgcp dtmf-relay** for all codecs.
Step 10	`Router(config)# mgcp modem passthru {cisco | ca}`	(Optional) Configures the gateway for modem and fax data.The keywords and arguments are as follows: • **cisco**—Specifies switching to the G.711 codec when the gateway detects a modem or fax tone to allow the analog data to pass-through. • **ca** (default)—Specifies switching to the CA that switches to G.711 codec when the gateway detects a modem or fax tone to allow the analog data to pass through. The **no** form of the command disables support for modem and fax data.
Step 11	`Router(config)# mgcp sdp simple`	(Optional) Specifies use of a subset of the session description protocol (SDP). Some call agents require this subset to send data through the network. The default is **no mgcp sdp simple**.

Configuring a TGW for SGCP

To configure a TGW for SGCP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mgcp`	Initiates the MGCP application.
Step 2	`Router(config)# mgcp call-agent` [*ipaddr* | *hostname*] [*port*] `service-type sgcp`	Specifies the call agent's IP address or domain name, the port, and gateway control service type. The keywords and arguments are as follows: • *ipaddr*—Specifies the call agent's IP address. • *hostname*—Specifies the call agent's hostname, using the format host.domain.ext. • *port*—Specifies the port for the call agent to use. Valid values are from 1025 through 65535. • **service-type**—Specifies the type of gateway control service supported by the call agent. Valid values are **mgcp** and **sgcp**. For SGCP configurations, use **sgcp**.
Step 3	`Router(config)# controller t1` *number*	Specifies the channel number of the T1 trunk to be used for analog calls.
Step 4	`Router(config-controller)# ds0-group` *channel-number* `timeslots` *range* `type` {`none` | `fgdos`} [*tone_type*] [*addr_info*] `service` {`sgcp` | `voice`}	Configures the channelized T1 time slots to accept the analog calls. For type **none**, use **service sgcp**. For type **fgdos**, use **service voice**. The keywords and arguments are as follows: • *channel-number*—Specifies the DS0 group number. Valid values are 1 to 23 for T1 interfaces and from 0 to 30 for E1 interfaces. • **timeslots** *range*—Specifies the DS0 time slot range of values. Valid values are from 1 to 24 for T1 interfaces and from 1 to 31 for E1 interfaces. The default value is 24. • **type**—Specifies the signaling type to be applied to the selected group. For SGCP functionality, use **none** or **fgdos**. • *tone_type*—Specifies the tone type supported by the signaling type. For signaling type **fgdos**, the valid value is **mf**. This parameter is available if **type** is **fgdos**. • *addr_info*—Specifies that calling and called party numbers are used. For type **fgdos**, the valid value is **dnis-ani**. This parameter is available if **type** is **fgdos**. • **service**—Specifies the type of service on the gateway. For SGCP configurations, valid values are **sgcp** or **voice**. For signaling type **none**, use **sgcp**. For signaling type **fgdos**, use **voice**.

Configuring an RGW

To configure an RGW, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mgcp`	Initiates the MGCP application. **Note** RGWs are configured only with MGCP.
Step 2	`Router(config)# mgcp call-agent` [*ipaddr* | *hostname*] [*port*] `service-type mgcp`	Specifies the call agent IP address or domain name, the port, and gateway control service type. The keywords and arguments are as follows: • *ipaddr*—Specifies the call agent's IP address. • *hostname*—Specifies the call agent's hostname, using the format host.domain.ext. • *port*—Specifies the port for the call agent to use. Valid values are from 1025 through 65535. • **service-type**—Specifies the type of gateway control service supported by the call agent. Valid values are **mgcp** or **sgcp**. For MGCP configurations, use **mgcp**.
Step 3	`Router(config)# dial-peer voice` *number* `pots`	Sets up the dial peer for a voice port.
Step 4	`Router(config-dial-peer)# application MGCPAPP`	Selects the MGCP application to run on the voice port.
Step 5	`Router(config-dial-peer)# exit`	Exits dial peer configuration mode and returns to global configuration mode.
Step 6	`Router(config)# mgcp package-capability` {`line-package` | `dtmf-package` | `gm-package` | `rtp-package`}	(Optional) Specifies the event packages that are supported on the gateway. The set of supported packages varies with the type of gateway (TGW or RGW). The keywords are as follows: • **line-package** (default)—Specifies the line package. • **dtmf-package**—Specifies the DTMF package. • **gm-package**—Specifies the generic media package. • **rtp-package**—Specifies the RTP package.
Step 7	`Router(config)# mgcp default-package` [`line-package` | `dtmf-package` | `gm-package`]	(Optional) Specifies the event package that should act as the default. Overrides the **mgcp package-capability** command.

Configuring the Cisco Voice Gateway 200 to Support Cisco CallManager

The Cisco Voice Gateway 200 functions as an RGW and uses the configuration steps shown in the Configuring an RGW section. In addition, the Cisco Voice Gateway 200 has the capability of using MGCP with Cisco CallManager for administration and redundant call agent features. This capability requires additional configuration steps.

To configure the Cisco Voice Gateway 200 so it can be controlled by Cisco CallManager using MGCP, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ccm-manager MGCP`	Enables support for Cisco CallManager within MGCP.
Step 2	`Router(config)# ccm-manager redundant host` *hostname1 hostname2*	Identifies one or two backup Cisco CallManager servers. The arguments *hostname1* and *hostname2* specify the first and second backup servers, respectively, using the dotted decimal format.
Step 3	`Router(config)# ccm-manager switchback` {**graceful** | **immediate** | **schedule-time** *hh:mm* | **uptime-delay** *minutes*}	Specifies how the gateway behaves if the primary server becomes unavailable and later becomes available again. The keywords and arguments are as follows: • **graceful**—Completes all outstanding calls before returning the gateway to the control of the primary Cisco CallManager server. • **immediate**—Returns the gateway to the control of the primary Cisco CallManager server without delay, as soon as the network connection to the server is reestablished. • **schedule-time** *hh:mm*—Returns the gateway to the control of the primary Cisco CallManager server at the specified time, where hh:mm is the time according to a 24-hour clock. If the gateway reestablishes a network connection to the primary server after the configured time, the switchback will occur at the specified time on the following day. • **uptime-delay** *minutes*—Returns the gateway to the control of the primary Cisco CallManager server when the primary server runs for a specified number of minutes after a network connection is reestablished to the primary server. Valid values are from 1 to 1440 (from 1 minute to 24 hours).

To force the Cisco Voice Gateway 200 to use the backup Cisco CallManager server, use the following command in privileged EXEC mode:

Command	Purpose
`Router# ccm-manager switchover-to-backup`	Redirects the Cisco Voice Gateway 200 gateway to the backup Cisco CallManager server.

Verifying the TGW or RGW Configuration

To verify the configuration settings for all platforms and protocols, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# show running-config`	Displays the current configuration settings.
Step 2	`Router(config)# show ccm-manager`	Displays the current configuration settings on the Cisco Voice Gateway 200.

Blocking New Calls and Gracefully Terminating Existing Calls

You can block all new MGCP calls to the router and gracefully terminate all existing active calls, which means that an active call is not terminated until the caller hangs up. To block all new calls, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mgcp block-newcalls`	Prevents the gateway from accepting new calls.
Step 2	`Router(config)# no mgcp block-newcalls`	Restarts normal MGCP call operation.

Monitoring and Maintaining MGCP

To monitor the MGCP configuration, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# show mgcp [connection | endpoint | statistics]`	Displays all active MGCP connections on the router.
Step 2	`Router# debug mgcp [all | errors | events | packets | parser]`	Turns on debugging for the gateway.
Step 3	`Router# clear mgcp statistics`	Resets the MGCP statistical counters.

MGCP Configuration Examples

This section provides configuration examples for each of the supported platforms:

Configuring the Cisco AS5300 As a TGW with MGCP Example

The following example illustrates a configuration only for MGCP calls. FGD-OS calls are not supported.

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname A
!
resource-pool disable
!
ip subnet-zero
ip ftp username smith
ip host B 209.165.200.225
ip host C 209.165.200.226
ip domain-name cisco.com
ip name-server 209.165.202.129
!
mgcp
mgcp request timeout 10000
mgcp call-agent 192.168.10.10 2302
mgcp restart-delay 5
mgcp package-capability gm-package
mgcp package-capability dtmf-package
mgcp package-capability trunk-package
mgcp package-capability rtp-package
mgcp package-capability as-package
mgcp package-capability mf-package
mgcp package-capability script-package
mgcp default-package trunk-package
mta receive maximum-recipients 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 1
 framing esf
 clock source line secondary 1
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
```

```
controller T1 2
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 3
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
voice-port 0:0
!
voice-port 1:0
!
voice-port 2:0
!
voice-port 3:0
!
interface Ethernet0
 ip address 192.168.10.9 255.255.255.0
 no ip directed-broadcast
!
interface FastEthernet0
 ip address 172.22.91.73 255.255.255.0
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
no ip classless
ip route 0.0.0.0 0.0.0.0 172.22.91.1
ip route 209.165.200.225 255.255.255.255 192.168.0.1
no ip http server
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Configuring the Cisco AS5300 As a TGW with SGCP Example

The following example illustrates a configuration that supports MGCP and FGD-OS calls:

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname A
!
resource-pool disable
!
ip subnet-zero
ip ftp username smith
ip host B 209.165.200.225
ip host C 209.165.200.226
ip domain-name cisco.com
ip name-server 209.165.202.129
```

```
!
mgcp
mgcp request timeout 10000
mgcp call-agent 192.168.10.10 2302 sgcp
mta receive maximum-recipients 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 1
 framing esf
 clock source line secondary 1
 linecode b8zs
 ds0-group 0 timeslots 1-24 type fgd-os mf dnis-ani service voice
!
controller T1 2
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 3
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
!voice-port 0:0
!
voice-port 1:0
!
voice-port 2:0
!
voice-port 3:0
!
interface Ethernet0
 ip address 192.168.10.9 255.255.255.0
 no ip directed-broadcast
!
interface FastEthernet0
 ip address 172.22.91.73 255.255.255.0
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
no ip classless
ip route 0.0.0.0 0.0.0.0 172.22.91.1
ip route 209.165.200.225 255.255.255.255 192.168.0.1
no ip http server
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Configuring the Cisco 3660 As a TGW with MGCP Example

The following example illustrates a platform that does not support FGD-OS calls.

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname A
!
memory-size iomem 40
voice-card 1
!
ip subnet-zero
!
mgcp 4000
mgcp call-agent 209.165.202.129 4000
mgcp package-capability gm-package
mgcp package-capability dtmf-package
mgcp package-capability rtp-package
mgcp package-capability as-package
isdn voice-call-failure 0
cns event-service server
!
controller T1 1/0
 framing esf
 clock source internal
 ds0-group 1 timeslots 1-24 type none service mgcp
!
controller T1 1/1
 framing esf
 clock source internal
 ds0-group 1 timeslots 1-24 type none service mgcp
!
voice-port 1/0:1
!
voice-port 1/1:1
!
interface FastEthernet0/0
 ip address 209.165.202.140 255.255.255.0
 no ip directed-broadcast
 load-interval 30
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 load-interval 30
 shutdown
 duplex auto
 speed auto
!
ip default-gateway 209.165.202.130
ip classless
ip route 209.165.200.225 255.255.255.255 FastEthernet0/0
no ip http server
!
snmp-server engineID local 00000090920000107BD8CD80
snmp-server community public RO
!
```

```
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Configuring the Cisco uBR924 As an RGW Example

The following example illustrates a platform that does not support FGD-OS calls.

```
version 12.2
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname A
!
logging buffered 200000 debugging
!
clock timezone - -8
ip subnet-zero
no ip routing
no ip domain-lookup
ip host A 192.168.147.91
ip host C 209.165.200.224
ip host D 209.165.200.225
!
mgcp
mgcp call-agent 192.168.10.10 2490
mgcp package-capability gm-package
mgcp package-capability dtmf-package
mgcp package-capability line-package
mgcp default-package line-package
!
voice-port 0
 input gain -3
!
voice-port 1
 input gain -3
!
dial-peer voice 1 pots
 application MGCPAPP
 port 1
!
dial-peer voice 2 pots
 application MGCPAPP
 port 0
!
interface Ethernet0
 ip address 192.168.147.91 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface cable-modem0
 ip address negotiated
 no ip directed-broadcast
 no ip route-cache
```

```
 no ip mroute-cache
 cable-modem downstream saved channel 459000000 20
cable-modem downstream saved channel 699000000 19 2
 cable-modem mac-timer t2 100000
 no cable-modem compliant bridge
 bridge-group 59
 bridge-group 59 spanning-disabled
!
ip default-gateway 10.1.1.1
ip classless
no ip http server
!
line con 0
 exec-timeout 0 0
 transport input none
line vty 0 4
 login
!
end
```

Configuring the Cisco 2620 As an RGW Example

The following example illustrates a platform that does not support FGD-OS calls.

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname D
!
memory-size iomem 10
ip subnet-zero
!
mgcp
mgcp call-agent 172.20.5.20
mgcp package-capability gm-package
mgcp package-capability dtmf-package
mgcp package-capability line-package
mgcp package-capability rtp-package
mgcp default-package line-package
cns event-service server
!
voice-port 1/0/0
!
voice-port 1/0/1
!
dial-peer voice 1 pots
 application MGCPAPP
 port 1/0/0
!
dial-peer voice 2 pots
 application MGCPAPP
 port 1/0/1
!
interface Ethernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Serial0/0
 no ip address
```

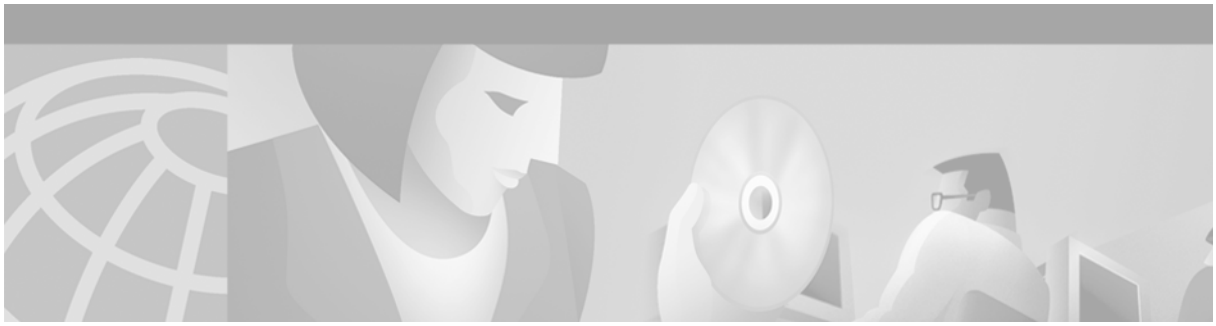
```
 no ip directed-broadcast
 no ip mroute-cache
 shutdown
 no fair-queue
!
interface Ethernet0/1
 ip address 172.20.5.25 255.255.255.0
 no ip directed-broadcast
!
interface Serial0/1
 no ip address
 no ip directed-broadcast
 shutdown
!
ip default-gateway 209.165.202.130
ip classless
ip route 209.165.200.225 255.255.255.224 Ethernet0/1
no ip http server
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Configuring the Cisco Voice Gateway 200 As an RGW Example

The following example illustrates the configuration of the Cisco Voice Gateway 200 as an RGW.

```
version 12.2
no service single-slot-reload-enable
no service pad
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
!
hostname MainVG200
!
ip subnet-zero
no ip finger
ip host dirt 172.16.1.129
!
mgcp
mgcp call-agent 172.20.71.44
call rsvp-sync
!
ccm-manager switchback immediate
ccm-manager redundant-host 172.20.71.47
ccm-manager mgcp
!
interface FastEthernet0/0
  ip address 172.21.10.14 255.255.255.0
  duplex auto
  speed auto
!
ip default-gateway 172.21.10.1
ip classless
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
ip route 172.16.0.0 255.255.0.0 172.20.82.1
```

```
no ip http server
!
access-list 199 permit udp any any range 16384 32766
access-list 199 permit ip host 10.51.26.6 any
access-list 199 permit ip host 10.51.16.7 any
queue-list 2 protocol ip 2 list 199
queue-list 2 default 5
queue-list 2 queue 2 byte-count 2880 limit 16
queue-list 2 queue 5 limit 1
priority-list 1 protocol ip high list 199
priority-list 1 default low
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
dial-peer voice 1 pots
!
dial-peer voice 111 pots
  application mgcpapp
  port 1/1/1
!
gateway
!
line con 0
  exec-timeout 0 0
  transport input none
line aux 0
line vty 0 4
  password lab
  login
!
end
```



H.323 Applications

This chapter provides an overview of the H.323 standard from the International Telecommunication Union Telecommunication Standardization Sector (ITU-T), of the Cisco H.323-compliant gatekeeper, of the Cisco H.323-compliant gateway, and of the Cisco H.323-compliant features. Cisco IOS software complies with the mandatory requirements and several of the optional features of the H.323 Version 2 specification. The chapter contains the following sections:

Refer to the ITU-T H.323 standard for more in-depth information about the overall H.323 standard.

For a complete description and for examples of configuring Cisco gatekeepers, see the chapter "Configuring H.323 Gatekeepers and Proxies."

For a complete description and for examples of configuring Cisco gateways, see the chapter "Configuring H.323 Gateways."

For more information on configuring Cisco H.323 features, see the "MGCP and Related Protocols," "Configuring SIP," "Voice over IP Overview," and "Dial Plans, Dial Peers, and Digit Manipulation" chapters. For general information regarding the H.323 Standard, refer to the ITU-T H.323 specifications.

For a more complete description of the H.323-compliant gatekeeper and H.323 Version 2 standard support upgrade commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index or search online.

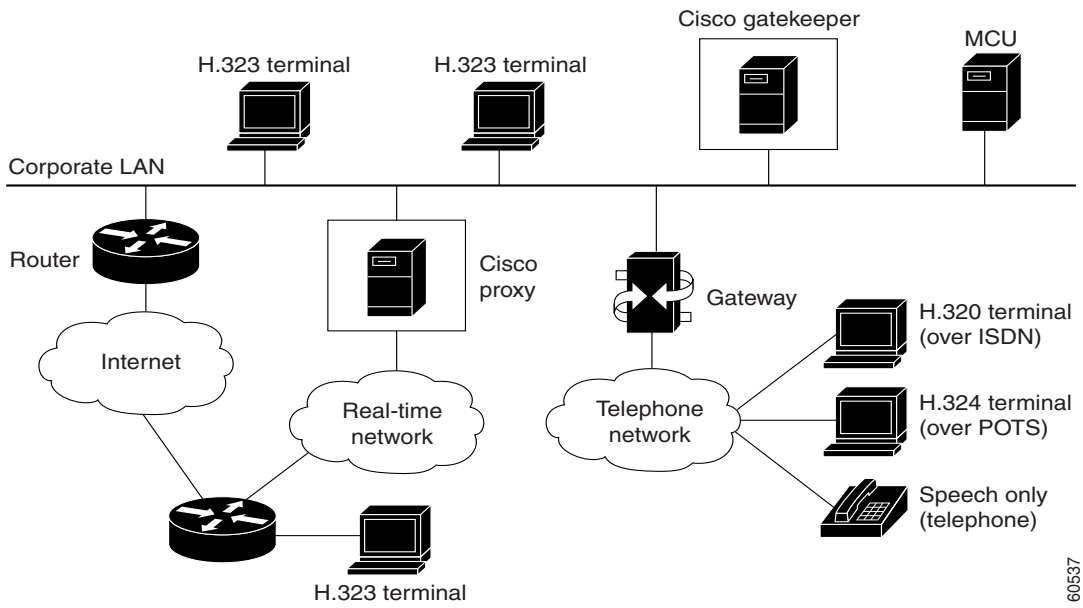
To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

The H.323 Standard

The H.323 standard provides for sending and receiving audio, video, and data on an IP-based internetwork. The following sections provide a basic overview of network components and how they relate to each other:

Figure 56 shows a typical H.323 network.

Figure 56 Gatekeeper in an H.323 Network



H.323 Terminals

An H.323 terminal is an endpoint in the network that provides for real-time, two-way communications with another H.323 terminal, gateway, or multipoint control unit (MCU). The communications consist of control, indications, audio, moving color video pictures, or data between the two terminals. A terminal may provide audio only; audio and data; audio and video; or audio, data, and video. The terminal can be a computer-based video conferencing system or other device.

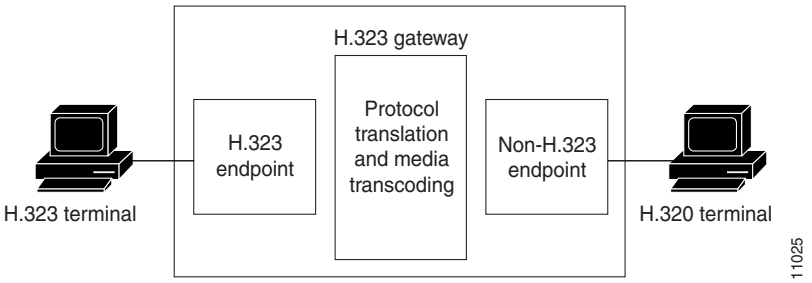
A gatekeeper supports a broad variety of H.323 terminal implementations from many different vendors. These terminals must support the standard H.323 Registration, Admission, and Status (RAS) protocol to function with the gatekeeper.

H.323 Gateways

An H.323 gateway is an endpoint on the LAN that provides real-time communications between H.323 terminals on the LAN and other ITU terminals on a WAN or to other H.323 gateways.

Gateways allow H.323 terminals to communicate with devices that are running other protocols. They provide protocol conversion between the devices that are running different types of protocols. For example, Figure 57 shows a gateway between an H.323 terminal and a non-H.323 terminal.

Figure 57 Gateway Between an H.323 Terminal and an H.320 Terminal



Configuring ISDN Redirect Number Support

Voice over IP (VoIP) supports the redirecting call feature of the VoIP gateway for ISDN calls. The redirecting number is an optional field of the Q.931 setup message.

When a local exchange carrier (LEC) switch detects an incoming call that is destined for a busy or nonanswering party, the switch formulates a Q.931 setup message with the redirecting number field set to the original destination number and sends it to the gateway. The called party number of the setup message will be set to one of the destination number (Dialed Number Identification Service [DNIS]) access numbers of the gateway.

If a redirect number is present on an incoming call, it is used in place of the DNIS. To configure ISDN redirect number support, see the “Dial Plans, Dial Peers, and Digit Manipulation” chapter.

H.323 Proxies

H.323 proxies are special types of gateways that relay H.323 calls to another H.323 endpoint. They can be used to isolate sections of an H.323 network for security purposes, to manage quality of service (QoS), or to perform special application-specific routing tasks.

H.323 Gatekeepers

An H.323 gatekeeper is an H.323 entity on the LAN that provides address translation and that controls access to the LAN for H.323 terminals, gateways, and MCUs.

Gatekeepers are optional nodes that manage endpoints in an H.323 network. The endpoints communicate with the gatekeeper using the RAS protocol.

Endpoints attempt to register with a gatekeeper on startup. When they wish to communicate with another endpoint, they request admission to initiate a call using a symbolic alias for the endpoint, such as an E.164 address or an e-mail address. If the gatekeeper decides that the call can proceed, it returns a destination IP address to the originating endpoint. This IP address may not be the actual address of the destination endpoint, but it may be an intermediate address, such as the address of a proxy or a gatekeeper that routes call signaling.

Note Although the gatekeeper is an optional H.323 component, it must be included in the network if proxies are used.

Gatekeeper Zones

An H.323 endpoint is an H.323 terminal, gateway, or MCU. An endpoint can call and be called.

H.323 endpoints are grouped into zones. Each zone has one gatekeeper that manages all the endpoints in the zone. A zone is an administrative convenience similar to a Domain Name System (DNS) domain. (Because a zone is, by definition, the area of control of a gatekeeper, you will find the terms “zone name” and “gatekeeper name” used synonymously in this chapter.)

Note The maximum number of local zones defined in a gatekeeper should not exceed 100.

MCUs

An MCU is an endpoint on the network that allows three or more endpoints to participate in a multipoint conference. It controls and mixes video, audio, and data from endpoints to create a robust multimedia conference. An MCU may also connect two endpoints in a point-to-point conference, which may later develop into a multipoint conference.

Note Some terminals have limited multipoint control built into them. These terminals may not require an MCU that includes all the functionality mentioned.

How Terminals, Gatekeepers, and Proxies Work Together

When endpoints are brought online, they first attempt to discover their gatekeeper. They discover their gatekeeper either by sending multicast a discovery request or by being configured with the address and, optionally, with the name of the gatekeeper and by sending a unicast discovery request. Following successful discovery, each endpoint registers with the gatekeeper. The gatekeeper keeps track of which endpoints are online and available to receive calls.

There are three ways to set up calls between various endpoints, as described in the following sections:

- Interzone Call with Proxy, page 240
- Interzone Call Without Proxy, page 239
- Interzone Call with Proxy, page 240

Intrazone Call

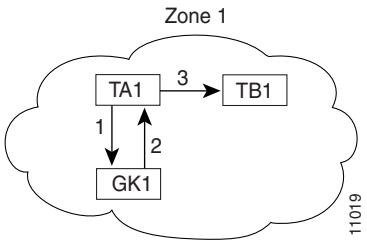
Intrazone calls occur within the same zone.

If terminal TA1 wants to make an intrazone call to terminal TB1 in Zone 1, the following sequence of events occurs:

1. TA1 asks GK1 for permission to call TB1.
2. GK1 returns the address of TB1 to TA1.
3. TA1 then calls TB1.

Figure 58 illustrates these events.

Figure 58 Intrazone Call



Interzone Call Without Proxy

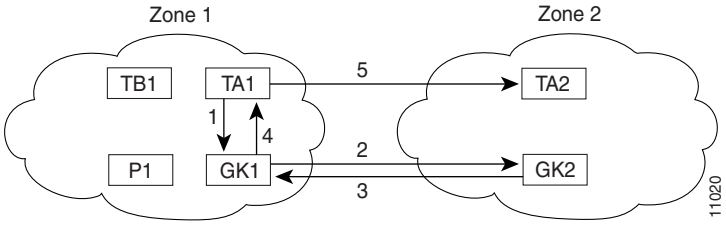
Interzone calls occur between two or more zones.

If terminal TA1 in Zone 1 wants to call terminal TA2 in Zone 2 without the use of a proxy, the following sequence of events occurs:

1. TA1 asks GK1 for permission to call TA2.
2. TA2 is not in the GK1 zone. GK1 locates GK2 as the TA2 gatekeeper. GK1 then asks GK2 for the TA2 address.
3. GK2 returns the TA2 address to GK1.
4. GK1 returns the address to TA1.
5. TA1 calls TA2.

Figure 59 illustrates these events.

Figure 59 Interzone Call Without Proxy



Interzone Call with Proxy

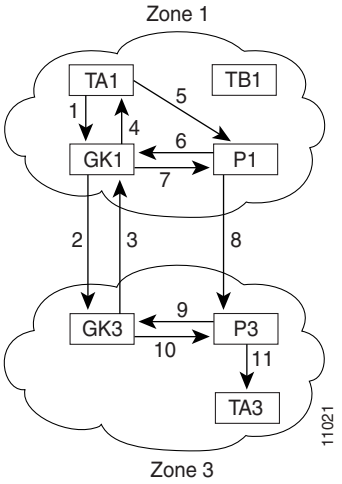
One reason for using a proxy is to isolate addressing information in one zone from another. When such isolation is desired, zones are configured as inaccessible on the gatekeepers. (Other reasons for using proxies are discussed later in this document.)

If terminal TA1 in Zone 1 wants to call terminal TA3 in Zone 3, the following sequence of events occurs:

1. TA1 asks GK1 for permission to call TA3.
2. GK1 locates GK3 as the TA3 gatekeeper. GK1 asks GK3 for the TA3 address.
3. GK3 responds with the P3 address instead of the TA3 address, to hide the TA3 identity.
4. GK1 knows that to get to P3, the call must go through P1. So GK1 returns the P1 address to TA1.
5. TA1 calls P1.
6. P1 consults GK1 to discover the true destination of the call (which is TA3 in this example).
7. GK1 instructs P1 to call P3.
8. P1 calls P3.
9. P3 consults GK3 for the true destination, which is TA3.
10. GK3 gives the TA3 address to P3.
11. P3 completes the call to TA3.

Figure 60 illustrates these events.

Figure 60 Interzone Call with Proxy



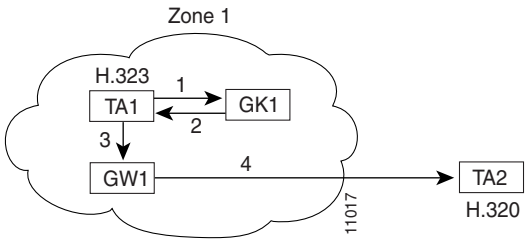
How Terminals, Gatekeepers, and Gateways Work Together

Gateways provide protocol conversion between terminals that run different types of protocols. Gateways communicate with gatekeepers using the RAS protocol. The gatekeeper maintains resource availability information, which it uses to select the appropriate gateway during the admission of a call. In Figure 61, the following conditions exist:

- TA1 is an H.323 terminal that is registered to GK1.
- GW1 is an H.323-to-H.320 gateway that is registered to GK1.
- TA2 is an H.320 terminal.

Figure 61 illustrates these events.

Figure 61 Intrazone Call Through Gateway



A call from TA1 to TA2 is set up as follows:

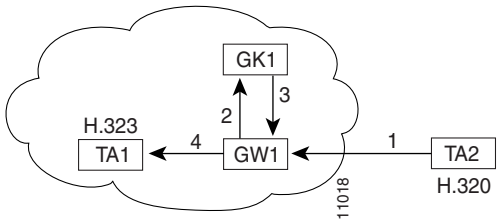
1. TA1 asks GK1 for permission to connect to the TA2 E.164 address.
2. The gatekeeper looks through its local registrations and does not find any H.323 terminals that are registered with that E.164 address, so the gatekeeper assumes that it is an H.320 terminal that is outside the scope of H.323. The gatekeeper instructs TA1 to connect to the GW1 IP address.
3. TA1 connects to GW1.
4. GW1 completes the call to TA2.

A call from TA2 to TA1 is set up as follows:

1. TA2 calls GW1 and provides the TA1 E.164 address as the final destination.
2. GW1 sends a message to GK1 asking to connect to that address.
3. GK1 gives GW1 the address of TA1.
4. GW1 completes the call with TA1.

Figure 62 illustrates these events.

Figure 62 Gateways Provide Translation Between Terminal Types



How Terminals, Gatekeepers, Proxies, and MCUs Work Together

When MCUs are brought online, they first attempt to discover their gatekeeper. As with terminals and proxies, MCUs discover their gatekeeper either by multicasting a discovery request or by being configured with the name and address of the gatekeeper and unicasting a discovery request. Following successful discovery, the MCU registers with the gatekeeper. The gatekeeper keeps track of which endpoints are online and available to receive calls.

There are three ways to set up an MCU conference call, as described in the following sections:

- Intrazone MCU Conference Call, page 243
- Interzone MCU Conference Call Without Proxy, page 243
- Interzone MCU Conference Call with Proxy, page 244

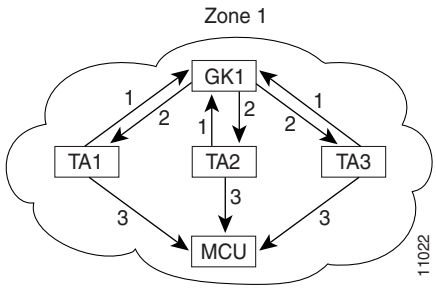
Intrazone MCU Conference Call

An MCU conference in Zone 1 is created with the conference ID CompanyMeeting. The MCU reregisters itself with the gatekeeper, with the new conference ID appended to its list of existing aliases. If terminals TA1, TA2, and TA3 in Zone 1 want to join CompanyMeeting, the following sequence of events occurs:

1. TA1, TA2, and TA3 join the conference by asking GK1 for permission to call the given conference ID.
2. GK1 returns the address of the MCU to TA1, TA2, and TA3.
3. TA1, TA2, and TA3 then call the MCU.

Figure 63 illustrates these events.

Figure 63 Intrazone Call with MCU



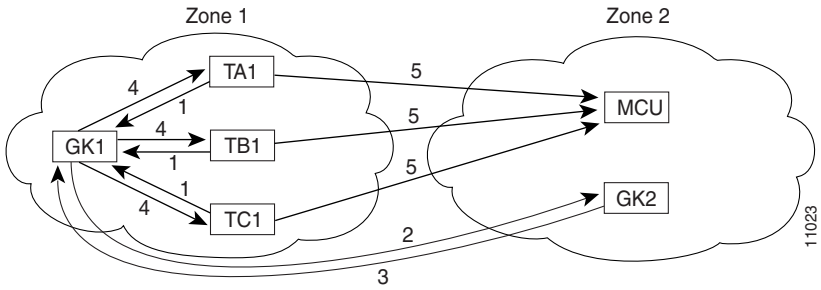
Interzone MCU Conference Call Without Proxy

The MCU in Zone 2 creates a conference with conference ID CompanyMeeting@zone2.com. The MCU reregisters itself with GK2, with the new conference ID appended to its list of existing aliases. Terminals TA1, TB1, and TC1 in Zone 1 want to join the MCU conference call with the conference ID CompanyMeeting@zone2.com in Zone 2. The following sequence of events occurs:

1. TA1, TB1, and TC1 ask GK1 for permission to join the conference.
2. GK1 locates GK2 for the remote zone that contains conference CompanyMeeting@zone2.com using DNS or information configured on GK1. GK1 sends a request to GK2 to recover the MCU address.
3. GK2 gives the MCU address to GK1.
4. GK1 gives the MCU address to TA1, TB1, and TC1, and it instructs these endpoints to set up the call with the MCU.
5. TA1, TB1, and TC1 then call the MCU.

Figure 64 illustrates these events.

Figure 64 Interzone MCU Conference Call Without Proxies



Interzone MCU Conference Call with Proxy

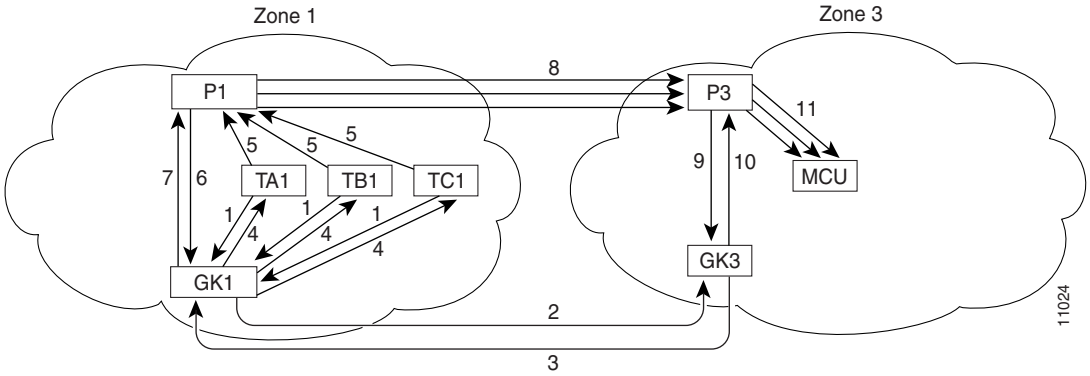
One main reason for using a proxy is to isolate addressing information in one zone from another. When such isolation is desired, zones are configured to be inaccessible on the gatekeepers.

The MCU in Zone 3 creates a conference with the conference ID CompanyMeeting@zone3.com. The MCU reregisters itself with the gatekeeper, using the new conference ID appended to its list of existing aliases. Terminals TA1, TB1, and TC1 in Zone 1 want to join the MCU conference with the conference ID CompanyMeeting@zone3.com in Zone 3. The following sequence of events occurs:

1. TA1, TB1, and TC1 ask GK1 for permission to join the conference CompanyMeeting@zone3.com.
2. GK1 locates GK3 for the remote zone that contains conference CompanyMeeting@zone3.com. GK1 asks GK3 for the MCU address.
3. GK3 responds with the PX3 address instead of the MCU address. GK1 knows that to get to PX3 the call should go through P1.
4. GK1 gives the P1 address to TA1, TB1, and TC1.
5. TA1, TB1, and TC1 call P1.
6. P1 consults GK1 to discover the true call destination, which is CompanyMeeting@zone3.com in this example.
7. GK1 instructs P1 to call P3.
8. P1 calls P3.
9. P3 consults with GK3 to discover the true call destination, which is CompanyMeeting@zone3.com in this example.
10. GK3 gives the MCU address to PX3.
11. P3 completes the call with the MCU.

Figure 65 illustrates these events.

Figure 65 Interzone MCU Conference Call with Proxy



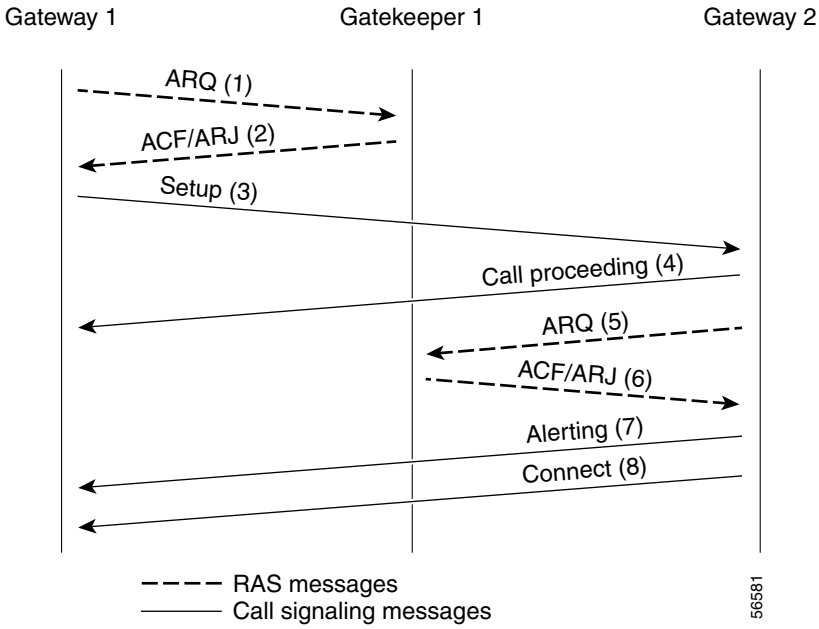
Call Signaling Procedures

Two important phases of H.323 call signaling are call setup and call termination. The following two examples demonstrate the call setup and call termination processes in relation to gatekeepers and gateways.

Call Setup—Both Gateways Registered to the Same Gatekeeper

In Figure 66, both gateways are registered to the same gatekeeper, and the gatekeeper has chosen direct call signaling. Gateway 1 (the calling gateway) initiates the admission request (ARQ) (1)/admission confirmation (ACF) (2) exchange with that gatekeeper. The gatekeeper returns the call signaling channel address of Gateway 2 (the called gateway) in the ACF. Gateway 1 then sends the setup (3) message to Gateway 2 using that transport address. If Gateway 2 wishes to accept the call, it initiates an ARQ (5)/ACF (6) exchange with the gatekeeper. Gateway 2 sends an alerting (7) message to Gateway 1. (If Gateway 2 receives an admission reject [ARJ] (6) message instead of an ACF message, it sends a release complete message to Gateway 1 instead of the alerting message.) Gateway 2 responds with the connect (8) message, which contains an H.245 control channel transport address for use in H.245 signaling.

Figure 66 Both Gateways Registered to the Same Gatekeeper



Call Termination

Either gateway may terminate a call in one of the following ways:

1. It discontinues transmission of video at the end of a complete picture and then closes all logical channels for video.
2. It discontinues transmission of data and then closes all logical channels for data.
3. It discontinues transmission of voice and then closes all logical channels for voice.
4. It transmits the H.245 endSessionCommand message in the H.245 control channel, indicating to the far end that it wishes to disconnect the call and then discontinues H.245 message transmission.
5. It waits to receive the endSessionCommand message from the other gateway and then closes the H.245 control channel.
6. If the call signaling channel is open, a release complete message is sent and the channel is closed.
7. The gateway clears the call by using the procedures defined below.

An endpoint receiving an endSessionCommand message without first having transmitted it carries out steps 1 and 7 above, except that in Step 5, the gateway waits for the endSessionCommand message from the first endpoint.

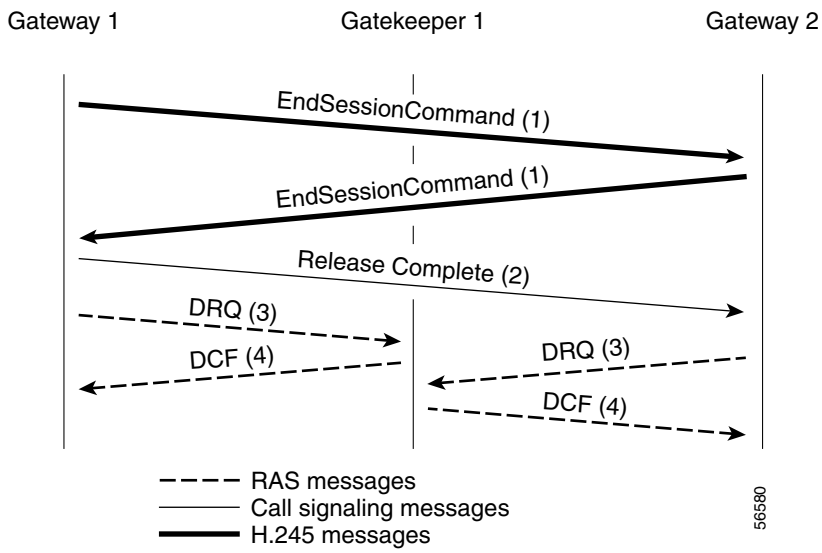
Terminating a call may not terminate a conference; a conference may be explicitly terminated using an H.245 message (**dropConference**). In this case, the gateways wait for the multipoint controller to terminate the calls as described.

Call Clearing with a Gatekeeper

In networks that contain a gatekeeper, the gatekeeper needs to know about the release of bandwidth. After performing steps 1 to 6 above, each endpoint transmits an H.225.0 disengage request (DRQ) message (3) to its gatekeeper (shown in Figure 67). The gatekeeper responds with a disengage confirm (DCF) message (4). After sending the DRQ message, the endpoints do not send further unsolicited information request response (IRR) messages that relate to that call to the gatekeeper. At this point, the call is terminated. Figure 67 shows the direct call model.

The DRQ and DCF messages are sent on the RAS channel.

Figure 67 Call Termination Direct Call Model



H.323 Feature Overview

This section includes the following subsections:

- Source Call Signal Address, page 248
- H.323 Version 2 Support, page 249
  - Lightweight Registration, page 250
  - Improved Gateway Selection Process, page 250
  - Gateway Resource Availability Reporting, page 251
  - Support for Single-Proxy Configurations, page 251
  - Registration of E.164 Addresses for Gateway-Attached Devices, page 251
  - Tunneling of Redirecting Number Information Element, page 251
  - DTMF Relay, page 252
  - H.245 Tunneling of DTMF Relay in Conjunction with Fast Connect, page 253
  - Translation of FXS Hookflash Relay, page 253

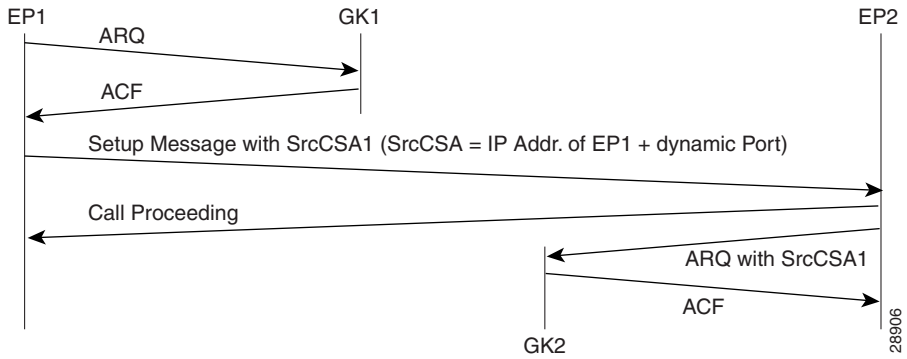
Source Call Signal Address

Source call signal address allows a source call-signal address field to be included in the ARQ.

Previously, in the Cisco IOS implementation of H.323 gateway software, if the terminating gateway was registered to an H.323 gatekeeper and used RAS, the ARQ message sent for each incoming call did not contain the H.225.0 source call signal address (CSA). The source CSA is an optional parameter in the ARQ message. The source CSA is also an optional parameter in the H.225.0 call setup message sent by the originating endpoint.

Source call signal address also allows for the source CSA parameter to be included in the ARQ message, as illustrated by the message sequence shown in Figure 68.

Figure 68 Source Call Signal Message Sequence



In the message sequence shown in Figure 68, the ARQ messages are enhanced to send the source CSA. The originating gateway (EP1) sends the H.225.0 setup message to the destination gateway. The setup message contains the source CSA parameter, which is the combination of the IP address of the originator and the dynamic TCP port number used or obtained for the H.225.0 call signaling channel. If the terminating gateway (EP2) accepts the call upon receipt of the setup message, the gateway sends an ARQ message to the gatekeeper. The terminating gateway retrieves the source CSA parameter sent by the originating gateway in the setup message. It then sends an ARQ message to the gatekeeper with the source CSA parameter. The CSA parameter is optional and has the same value as the source CSA in the received setup message. If the setup message does not contain the source CSA parameter, the terminating gateway determines the source CSA by using the H.225.0 call-signaling TCP socket connection of the peer endpoint, which it uses in the ARQ message.

If the originating gateway is registered to a gatekeeper and RAS is used as the session target, the originating gateway also sends an ARQ message. This ARQ does not include the optional source CSA parameter.

H.323 Version 2 Support

Cisco software complies with the mandatory requirements and several of the optional features of the H.323 Version 2 specification. Cisco H.323 Version 2 software enables gatekeepers, gateways, and proxies to send and receive all the required fields in H.323 Version 2 messages. Cisco H.323 Version 2 features include the following:

Lightweight Registration

Before the release of its H.323 Version 2 software, Cisco gateways reregistered with the gatekeeper every 30 seconds. Each registration renewal used the same process as the initial registration, even though the gateway was already registered with the gatekeeper. These registration renewals generated considerable overhead at the gatekeeper.

Cisco H.323 Version 2 software defines a lightweight registration procedure that still requires the full registration process for initial registration but that uses an abbreviated renewal procedure to update the gatekeeper and minimize overhead.

Lightweight registration requires each endpoint to specify a time-to-live (TTL) value in its registration request (RRQ) message. When a gatekeeper receives an RRQ message with a TTL value, it returns an updated TTL timer value in a registration confirmation (RCF) message to the endpoint. Shortly before the TTL timer expires, the endpoint sends an RRQ message with the KeepAlive field set to TRUE, which refreshes the existing registration.

It is not required that an H.323 Version 2 endpoint indicate a TTL in its registration request. If the endpoint does not indicate a TTL, the gatekeeper assigns one and sends it to the gateway in the RCF message. No configuration changes are permitted during a lightweight registration, so all fields other than the endpointIdentifier, gatekeeperIdentifier, tokens, and TTL are ignored. In the case of H.323 Version 1 endpoints that cannot process the TTL field in the RCF, the gatekeeper probes the endpoint with information requests (IRQs) for a predetermined grace period to see if the endpoint is still alive.

Improved Gateway Selection Process

Cisco H.323 Version 2 software improves the gateway selection process as follows:

- When more than one gateway is registered in a zone, the updated **zone prefix** command allows selection priorities to be assigned to these gateways on the basis of the dialed prefix.
- Gateway resource reporting allows the gateway to notify the gatekeeper when H.323 resources are getting low. The gatekeeper uses this information to determine which gateway it will use to complete a call.

The gatekeeper maintains a separate gateway list, ordered by priority, for each of its zone prefixes. If a gateway does not have an assigned priority for a zone prefix, it defaults to priority 5, which is the median. To explicitly bar the use of a gateway for a zone prefix, the gateway must be defined as having a priority 0 for that zone prefix.

When selecting gateways, the gatekeeper identifies a target pool of gateways by performing a longest zone prefix match; then it selects from the target pool according to priorities and resource availability. If all high-priority gateways are busy, a low-priority gateway might be selected.

Gateway Resource Availability Reporting

To allow gatekeepers to make intelligent call routing decisions, the gateway reports the status of its resource availability to its gatekeeper. Resources that are monitored are digital signal level 0 (DS0) channels and digital signal processor (DSP) channels. In Cisco IOS Release 12.1, this feature is available only on the AS5300 platform.

The gateway reports its resource status to the gatekeeper using the RAS Resource Availability Indication (RAI). When a monitored resource falls below a configurable threshold, the gateway sends a RAI to the gatekeeper indicating that the gateway is almost out of resources. When the available resources then cross over another configurable threshold, the gateway sends a RAI indicating that the resource depletion condition no longer exists. Resource reporting thresholds are configured by using the **resource threshold** command. The upper and lower thresholds are separately configurable to prevent the gateway from operating sporadically because of the availability or lack of resources.

Support for Single-Proxy Configurations

Cisco H.323 Version 2 software supports single-proxy, two-proxy, and no-proxy calls. Proxies can also be independently configured to meet the needs of inbound and outbound call scenarios.

Registration of E.164 Addresses for Gateway-Attached Devices

If phones are connected directly to the gateway, the Cisco H.323 Version 2 gateway allows fully qualified E.164 numbers to be registered with the gatekeeper. When configuring the gateway, use the **register** command to register these E.164 numbers.

Tunneling of Redirecting Number Information Element

An incoming PRI setup message may contain either a Redirecting Number (RDN) Information Element (IE) or an Original Called Number (OCN) IE. These IEs indicate that the call has been redirected (forwarded) and that each message contains the following:

- The destination number (DN) that was originally called
- The reason for the call being redirected
- Other related information

OCN IE is a Nortel variant of the RDN IE.

The H.323 Version 2 gateway passes the entire RDN or OCN IE from an incoming PRI message into the H.225.0 setup message. The IE is encapsulated in the nonStandardData field within the user-to-user information element (UUIE) of the H.225.0 setup message. The nonStandardData field can contain the encapsulated RDN or OCN IE and a tunneled global, signaling, and control standard QSIG message, or it can contain only the OCN or RDN. Cisco and other third-party H.323 endpoints can access the redirected information by decoding the nonStandardData field. In accordance with the H.225.0 specification, the nonStandardData is ignored by third-party endpoints and causes no interoperability problems.

For redirected PRI calls that are routed to a Cisco gateway, that are sent using H.323 to another Cisco gateway, and that exit the gateway using PRI, the RDN/OCN IE is tunneled from the source gateway to the destination gateway. The incoming PRI setup message is tunneled through H.225.0 and is encoded into the outgoing PRI setup message by the destination gateway.

Tunneling the RDN or OCN IE is important for applications such as Unified Messaging servers that need to know the telephone number that was originally dialed so as to access the correct account information.

DTMF Relay

Dual-Tone Multifrequency (DTMF) is the tone generated on a touchtone phone when the keypad digits are pressed. During a call, DTMF may be entered to access interactive voice response (IVR) systems, such as voice mail and automated banking services.

In previous releases of Cisco IOS software, DTMF is transported in the same way as voice. This approach can result in problems accessing IVR systems. Although DTMF is usually transported accurately when using high-bit-rate voice codecs such as G.711, low-bit-rate codecs such as G.729 and G.723.1 are highly optimized for voice patterns and tend to distort DTMF tones. As a result, IVR systems may not correctly recognize the tones.

DTMF relay solves the problem of DTMF distortion by transporting DTMF tones "out-of-band" or separate from the encoded voice stream. Cisco H.323 Version 2 software introduces the following three options to the existing **dtmf-relay** command for sending DTMF tones out-of-band:

- A Cisco proprietary RTP-based method (**dtmf-relay cisco-rtp** command)
- H.245 signal (**dtmf-relay h245-signal** command)
- H.245 alphanumeric (**dtmf-relay h245-alphanumeric** command)

If none of these options is selected, DTMF tones are transported in-band and encoded in the same way as voice traffic.

The **dtmf-relay cisco-rtp** command sends DTMF tones in the same Real-Time Protocol (RTP) channel as voice. However, the DTMF tones are encoded differently from the voice samples and are identified by a different RTP payload type code. This method accurately transports DTMF tones, but because it is proprietary, it requires the use of Cisco gateways at both the originating and terminating endpoints of the H.323 call.

The **dtmf-relay h245-signal** and **dtmf-relay h245-alphanumeric** commands are modes of DTMF transport defined by the ITU H.245 standard. These methods separate DTMF digits from the voice stream and send them through the H.245 signaling channel instead of the RTP channel. The tones are transported in H.245 user input indication messages. The H.245 signaling channel is a reliable channel, so the packets that transport the DTMF tones are guaranteed to be delivered. However, because of the overhead that is generated by using a reliable protocol, and depending on network congestion conditions, the DTMF tones may be slightly delayed. This delay is not known to cause problems with existing applications.

The **dtmf-relay h245-signal** command relays a more accurate representation of a DTMF digit than does the **dtmf-relay h245-alphanumeric** command because tone duration information is included along with the digit value. This information is important for applications requiring that a key be pressed for a particular length of time. For example, one popular calling card feature allows the caller to terminate an existing call by pressing the # key for more than 2 seconds and then making a second call without having to hang up in between. This feature is beneficial because the access number and personal identification number (PIN) code do not need to be dialed again. Outside-line access charges, which are common at hotels, may also be avoided.

The **dtmf-relay h245-alphanumeric** command simply relays DTMF tones as ASCII characters. For instance, the DTMF digit 1 is transported as the ASCII character 1. There is no duration information associated with tones in this mode. When the Cisco H.323 gateway receives a DTMF tone using this method, it will generate the tone on the Public Switched Telephone Network (PSTN) interface of the call using a fixed duration of 500 milliseconds. All systems that are H.323 Version 2-compliant are required to support the **dtmf-relay h245-alphanumeric** command, but support of the **dtmf-relay h245-signal** command is optional.

The ability of a gateway to receive DTMF digits in a particular format and the ability to send digits in that format are independent functions. No configuration is necessary to receive DTMF digits from another H.323 endpoint using any of the methods described. The Cisco H.323 Version 2 gateway is capable of receiving DTMF tones transported by any of these methods at all times.

However, to send digits out-of-band using one of these methods, two conditions must be met:

- The chosen method of DTMF relay must be enabled during dial-peer configuration using the **dtmf-relay** command.
- The peer (the other endpoint of the call) must indicate during call establishment that it is capable of receiving DTMF in that format.

More than one DTMF relay option may be enabled for a particular dial peer. If more than one option is enabled, and if the peer indicates that it is capable of receiving DTMF in more than one of these formats, the gateway will send DTMF using the method among the supported formats that it considers to be the most preferred. The preferences are defined as follows:

- **dtmf-relay cisco-rtp** (highest preference)
- **dtmf-relay h245-signal**
- **dtmf-relay h245-alphanumeric**

If the peer is not capable of receiving DTMF in any of the modes that were enabled, DTMF tones will be sent in-band.

When the Cisco H.323 Version 2 gateway is involved in a call to a Cisco gateway that is running a version of Cisco IOS software prior to Release 12.0(5)T, DTMF tones will be sent in-band because those systems do not support DTMF relay.

See the “Configuration Task List” section in the “Configuring H.323 Gateways and Proxies” chapter for an example of configuring DTMF relay.

H.245 Tunneling of DTMF Relay in Conjunction with Fast Connect

Through H.245 tunneling, H.245 messages are encapsulated within H.225.0 messages without using a separate H.245 TCP connection. When tunneling is enabled, one or more H.245 messages can be encapsulated in any H.225.0 message. H.245 tunneling is not supported as a stand-alone feature; initiation of H.245 tunneling procedures can be initiated only by using the **dtmf-relay** command and only from an active fast connect call. Furthermore, if **dtmf-relay** is configured on a Version 2 VoIP dial peer and the active call has been established by using fast connect, tunneling procedures initiated by the opposite endpoint are accepted and supported.

H.245 tunneling is backward compatible with H.323 Version 1 configurations.

Translation of FXS Hookflash Relay

A hookflash indication is a brief on-hook condition that occurs during a call. It is not long enough in duration to be interpreted as a signal to disconnect the call. Create a hookflash indication by quickly depressing and then releasing the hook on your telephone.

PBXs and telephone switches are frequently programmed to intercept hookflash indications and use them as a way to allow a user to invoke supplemental services. For example, your local service provider may allow you to enter a hookflash as a means of switching between calls if you subscribe to a call waiting service.

In the traditional telephone network, a hookflash results in a voltage change on the telephone line. Because there is no equivalent of this voltage change in an IP network, the ITU H.245 standard defines a message representing a hookflash. To send a hookflash indication using this message, an H.323 endpoint sends an H.245 user input indication message containing a "signal" structure with a value of "!". This value represents a hookflash indication.

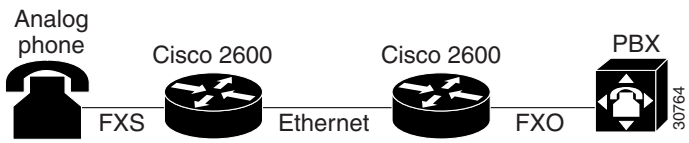
Cisco H.323 Version 2 software includes limited support for relaying hookflash indications using the H.245 protocol. H.245 user input indication messages containing hookflash indications that are received on the IP call leg are forwarded to the plain old telephone service (POTS) call leg if the POTS interface is Foreign Exchange Office (FXO). If the interface is not FXO, any H.245 hookflash indication that is received is ignored. This support allows IP telephony applications to send hookflash indications to a PBX through the Cisco gateway and thereby invoke the IOS supplementary services of the PBX if the PBX supports access to those features using hookflash.

The gateway does not originate H.245 hookflash indications in this release. For example, it does not forward hookflash indications from Foreign Exchange Station (FXS) interfaces to the IP network over H.245.

The acceptable duration of a hookflash indication varies by equipment vendor and by country. Although one PBX may consider a 250-millisecond on-hook condition to be a hookflash, another PBX may consider this condition to be a disconnect. Therefore, the **timing hookflash-out** command allows the administrator to define the duration of a hookflash signal generated on an FXO interface.

Figure 69 illustrates an FXS hookflash being translated to an H.245 user input.

Figure 69 Translating an FXS Hookflash to an H.245 User Input



In the Cisco H.323 Version 2 software, an FXS hookflash relay is generated only if the following two conditions are met:

- The other endpoint must support the reception of an H.245 hookflash and advertise this using the "Receive User Input Capability" message during H.245 capabilities exchange.
- The call must be established with either the **h245-alphanumeric** or **h245-signal** variant of the **dtmf-relay** command.

This implies that the VoIP dial peer must be configured for **dtmf-relay h245-alphanumeric** or **h245-signal**, but not **cisco-rtp**.

Enter the **timing hookflash-input** command on FXS interfaces to specify the maximum length in milliseconds of a hookflash indication. If the hookflash lasts longer than the specified limit, then the FXS interface processes the indication as an onhook.

The acceptable duration of a hookflash indication varies by equipment vendor and by country. One PBX may consider a 250 milliseconds on-hook condition to be a hookflash; another PBX may consider this condition to be a disconnect.

H.235 Security

Security for RAS protocol signaling between H.323 endpoints and gatekeepers is enhanced in H.323 Version 2 software by including secure endpoint registration of the Cisco gateway to the Cisco gatekeeper and secure per-call authentication. In addition, it provides for the protection of specific messages related to Open Settlement Protocol (OSP) and to other messages as required via encryption tokens. The authentication type is "password with hashing" as described in the ITU H.235 specifications. Specifically, the encryption method is to use the MD5 algorithm, with password hashing. This functionality is provided by the **security token required-for** command on the gatekeeper and the **security password** command on the gateway.

The gatekeeper can interact with a RADIUS security server to perform the authentications. The gateway can also authenticate an external application by using the Gatekeeper Transaction Message Protocol (GKTMP) application programming interface (API).

Per-call authentication is accomplished by validating account and pin numbers that are entered by the user connected to the calling gateway by using an IVR prompt.

The security mechanisms described above require the gateway and gatekeeper clocks to be synchronized within 30 seconds of each other by using a Network Time Protocol (NTP) server.

GKTMP and RAS Messages

The GKTMP for the Cisco gatekeeper provides a transaction-oriented application protocol that allows an external application to modify gatekeeper behavior by processing specified RAS messages.

A set of triggers can be specified that use RAS messages that can be recognized by the gatekeeper. Triggers are specified filter conditions that must match each type of RAS message. The triggers can be dynamically registered by using the external application, or this information can be configured by using the command-line interface (CLI) on the gatekeeper.

When the gatekeeper receives a RAS message that meets the specified trigger conditions, it forwards the message to the external application in a GKTMP message format. This message is text encoded and sent over TCP. The external application can then modify fields in the message before returning it to the gatekeeper for further processing, or it may return a RAS response to the gatekeeper to be forwarded to the RAS client.

The following messages can be sent in GKTMP:

- ACF—admission confirm
- ARJ—admission reject
- ARQ—admission request
- LCF—location confirm
- LRJ—location reject
- LRQ—location request
- RCF—registration confirm
- RRJ—registration reject
- RRQ—registration request
- URQ—unregistration request

The application server interprets RAS messages in the following ways:

- For RRQ and URQ, the application server performs gatekeeper authorization, storing endpoint RAS gatekeeper IP addresses and maintaining gatekeeper resource control.

- For ARQ and LRQ, the application server performs authorization and digit translation functions and returns terminating IP addresses or a new E.164 address to the gatekeeper for reorigination by the originating gateway.
- For LCF and LRJ, the application server intercepts location responses from a distant gatekeeper and modifies the message fields before responding to the originating gateway.

Note Cisco has developed an API that can be used to provide an interface to the Cisco gatekeeper. Refer to the *Cisco Gatekeeper External Interface Reference*.

To configure the gatekeeper to receive trigger registrations from the external applications, specify the registration port of the server using the **server registration-port** command. This command tells the gatekeeper to listen for server connections.

You can also configure the gatekeeper to initiate the connection to a specified external application by using the **server trigger** command to specify a set of static trigger conditions for a specified server. Only one application server can be specified for each **server trigger** command. All RAS messages that do not match the selection criteria for any external application are processed normally by the gatekeeper. The **show gatekeeper servers** and **debug gatekeeper servers** commands can be entered to assist in the configuration.

See the "Gatekeeper Transaction Message Protocol and RAS Messages Example" in the "Configuring H.323 Gatekeepers and Proxies" chapter of this configuration guide.

RAS Message Fields

In support of the H.323 security and accounting features, fields have been added to several of the RAS messages effective with Cisco IOS Release 12.0(7)T. In general, all the RAS messages sent by the gateway, with the exception of the gateway request (GRQ), include authentication data in the cryptoToken field. This section lists each of the messages that changed effective with Cisco IOS Release 12.0(7)T and describes the fields that have been added.

GRQ Message

When H.323 security is enabled on the gateway, the following fields are added to the GRQ message:

Field	Description
authenticationCapability	This field should have a value of pwdHash.
algorithmOIDs	The object ID for the MD5 algorithm. The object identifier (OID) used to indicate MD5 will be {1 2 840 113549 2 5}.

GCF Message

When H.323 security is enabled on the gateway, the following fields should be in the gateway confirmation (GCF) message:

Field	Description
authenticationMode	This field should have a value of pwdHash.
algorithmOIDs	The object ID for the MD5 algorithm. The OID used to indicate MD5 will be {1 2 840 113549 2 5}.

If the authenticationMode or the algorithm OIDs fields do not contain the values specified above, the gatekeeper responds with a gatekeeper rejection (GRJ) message that contains a reject reason of securityDenial. This prompts the gateway to resend the GRQ .

RRQ Message

If H.323 security is enabled on the gateway, the following fields are added to the RRQ message:

Field	Description
cryptoTokens	This field contains one of the cryptoToken types defined for the CryptoH323Token field specified in H.225.0. Currently, the only type of cryptoToken supported is cryptoEPPwdHash.

The following fields are contained within the cryptoEPPwdHash structure:

Field	Description
alias	The gateway alias, which is the H.323 ID of the gateway.
timestamp	The current time stamp.
token	The MD5 encoded PwdCertToken. This field contains the following: timestamp—The same as the time stamp of cryptoEPPwdHash. password—The password of the gateway. generalID—The same gateway alias as the one included in the cryptoEPPwdHash. tokenID—The object ID.

ARQ Message

When H.323 security is enabled on the gateway, additional fields are included in the ARQ message. The contents of the field depend on whether the ARQ message is sent from the source gateway or the destination gateway.

Source Gateway ARQ Message

If the ARQ message is sent from the source gateway, the following fields are included:

Field	Description
cryptoTokens	This field contains one of the cryptoToken types defined for the CryptoH323Token field specified in H.225.0. Currently, the only type of cryptoToken supported is cryptoEPPwdHash.

The following fields are contained within the cryptoEPPwdHash structure:

Field	Description
alias	The account number of the user or the H.323 ID of the gateway if endpoint authentication is selected.
timestamp	The current time stamp.
token	The MD5 encoded PwdCertToken. This field contains the following: • timestamp—The same as the time stamp of cryptoEPPwdHash. • password—If "endpoint" is selected, this is the security password of the gateway. Otherwise, it is the password or PIN of the user. • generalID—If "endpoint" is selected, this is the H.323 ID of the gateway. Otherwise, it is the ID or account number of the user. • tokenID—The object ID.

Destination Gateway ARQ Message

If the ARQ message is sent from the destination gateway, the following fields are included:

Field	Description
cryptoTokens	This field contains one of the cryptoToken types defined for the CryptoH323Token field specified in H.225.0. Currently, the only type of cryptoToken supported is cryptoEPPwdHash.

The following fields are contained within the cryptoEPPwdHash structure:

Field	Description
alias	The alias (H.323 ID or E.164 address) of the destination gateway.
timestamp	The current time stamp.

Field	Description
token	The MD5 encoded PwdCertToken. This field contains the following: timestamp—The same as the time stamp of cryptoEPPwdHash. password—The password of the destination gateway. generalID—The same gateway alias as the one included in cryptoEPPwdHash. tokenID—The object ID.

ACF Message

If H.323 security is enabled on the gateway, the gatekeeper should include the billing-related information from the nonStandardParameter field of the clearTokens structure. If the call is using a prepaid call service, the clearTokens field should indicate the maximum call duration. In the case of prepaid call service, the gateway will terminate the call if it exceeds the allowed time.

The following clearToken fields should be included in the ACF message:

Field	Description
nonStandard	The billing information for the call.
tokenOID	The generic billing object ID.

The following fields are contained within the nonStandardParameter structure:

Field	Description
nonStandardIdentifier	The generic billing object ID.
BillingInfo	The billing information. This field can contain the following: • bill_to—A string that identifies the subscriber that should be billed for this call. • reference_id—A unique ID generated by the billing system. • billing_mode—Whether the call is being made using prepaid call service (debit_mode) or not (credit_mode). • max_duration—The maximum duration allowed for the call. Used only for prepaid call service. • balance—The account balance of the caller. For a billing mode of credit_mode, this should be a negative value that represents the current amount owed by the subscriber. Otherwise, this should be a positive value that represents the credit remaining on the debit account of the subscriber. • currency—The currency used in reporting the balance. • timezone—The time zone of the call, represented by a hexadecimal string that indicates the difference in seconds between the location of the caller and the Universal Time Coordinated (UTC).

DRQ Message

The gateway sends a DRQ message when the call ends. If H.323 security is enabled on the gateway, the call usage information is included in the DRQ message. The call usage information is sent in the nonStandardParameter field of the ClearToken structure.

The following fields are contained within the nonStandardParameter structure:

Field	Description
duration	The duration of the call in seconds.
callLog	The call usage information. This field contains the following information: • DISCONNECT_REASON—The disconnect reason. Possible values are as follows: – DISCONNECT_NORMAL—The call ended normally. – DISCONNECT_DISCONNECT—The call ended because of a technical failure. – DISCONNECT_ABANDONED—The call never took place; for example, the remote phone was not answered. – DISCONNECT_PREEMPT—The call was ended by the gateway. This would be the disconnect reason issued if the call was ended because the max_duration was exceeded. • DISCONNECT_STRING—A string that further describes the disconnect reason. • TIME—The time at which the call started, indicated by a hexadecimal string that represents the time, in seconds, since 00:00 January 1, 1970 UTC. • ORIGIN—Whether the call was inbound or outbound.

Multizone Features

Cisco multizone software enables the Cisco gateway to provide information to the gatekeeper using additional fields in the RAS messages. The gatekeeper no longer terminates a call if it is unable to resolve the destination E.164 phone number with an IP address.

Previously, the source gateway attempted to set up a call to a destination IP address as provided by the gatekeeper in an admission confirm (ACF) message. If the gatekeeper was unable to resolve the destination E.164 phone number to an IP address, the incoming call was terminated.

Multizone software allows a gatekeeper to provide additional destination information and modify the destinationInfo field in the ACF message. The gateway will include the canMapAlias-associated destination information in setting up the call to the destination gateway.

The gatekeeper indicates to the gateway that the call should be destined to a new E.164 number by sending an ACF message with an IP address of 10.0.0.0 in the destCallSignalAddress field and the new destination E.164 phone number in the destinationInfo field.

The gateway that receives such an ACF will fall back to routing the call on the basis of this new E.164 address and performing a relookup of the configured dial plan for the gateway. If the gateway routes the call on the basis of the new E.164 address, the call might be routed back to the PSTN or to an H.323 endpoint.

Codec Negotiation

Codec negotiation allows the gateway to offer several codecs during the H.245 capability exchange phase and to ultimately settle on a single common codec during the call establishment phase. Offering several codecs increases the probability of establishing a connection because there will be a greater chance of overlapping voice capabilities between endpoints. Normally, only one codec can be specified when a dial peer is configured, but codec negotiation allows a prioritized list of codecs associated with a dial peer to be specified. During the call establishment phase the router will use the highest priority codec from the list that it has in common with the remote endpoint. It will also adjust to the codec selected by the remote endpoint so that a common codec is established for both the receive and send voice directions.

When a call is originated, all the codecs associated with the dial peer are sent to the terminating endpoint in the H.245 terminal capability set message. At the terminating endpoint, the gateway will advertise all the codecs that are available in firmware in its terminal capability set. If there is a need to limit the codecs advertised to a subset of the available codecs, a terminating dial peer must be matched that includes this subset. The **incoming called-number** command in dial peer configuration mode can be used to force this match.

Supported Codecs

The supported codecs are available for use with Cisco H.323 Version 2 software. Table 21 lists each codec with a default packet size (in bytes) and a range.

Table 21 Codec Default Packet Size

Codecs	Range (in bytes)	Default (in bytes)	Bit Rate
G.711ulaw	40–240	160	64 kbps
G.711alaw	40–240	160	64 kbps
G.723r63	24–240	24	6.3 kbps
G.723r53	20–240	20	5.3 kbps
G.723ar63	24–240	24	6.3 kbps
G.723ar53	20–240	20	5.3 kbps
G.726r32	20–240	40	32 kbps
G.726r24	15–240	30	24 kbps
G.726r16	10–240	20	16 kbps
G.728	10–240	10	16 kbps
G.729br8	10–240	20	8 kbps
G.729r8 pre-ietf	10–240	20	8 kbps
G.729r8	10–240	20	8 kbps

Note A separate codec for G.729 Annex B is included, which adds Annex B functionality to G.729. A separate codec for G.723.1 Annex A adds Annex A functionality to G.723.1.

Note The Annex B functionality added to G.729 and the Annex A functionality added to G.723.1 are the built-in, codec-specific voice-activated detection/calling tone (VAD/CNG) functions.

H.245 Empty Capabilities Set

Empty capabilities set support is a mandatory part of the H.323 Version 2 standard. It is used by applications to redirect the voice media stream. This feature is particularly useful for applications such as the following:

- Selsius IP phones, which rely on a hub or call manager to direct the media stream to IP phones.
- Unified messaging for which it is desirable to redirect the media stream to various message servers for message playout.

The empty capabilities set feature was added to provide a way to redirect RTP streams. The RTP streams are redirected as follows:

- The sequence starts with the an empty capabilities set being received at an endpoint.
- After an open logical channel (OLC) is established (or if in the middle of this process) one of the endpoints sends an empty capabilities set message.
- When the empty capabilities set message is received, the other endpoints close the logical channel if any was opened with that endpoint and move to a pause state, waiting for a nonempty capability set message.

After receiving the nonempty capabilities set message, the endpoint moves to the beginning of Phase B, which is the initial communication and capabilities exchange, as described in H.323 Version 3 (June 1999), item 8.4.6.

In other words, the exchange of the capabilities message determines a master/slave relationship, and a new OLC message is created to open a new logical channel with another endpoint. From this point on, the RTP streams are sent to the new endpoint.

H.323 Version 2 Fast Connect

Fast connect allows endpoints to establish media channels without waiting for a separate H.245 connection to be opened. This streamlines the number of messages that are exchanged and the amount of processing that must be done before endpoint connections can be established. A high-level view of the fast connect procedures within the H.323 protocol follows:

1. The calling endpoint transmits a setup message containing the fastStart element that contains a sequence of encoded logical channel structures, each representing a different capability media type for both “send” and “receive” directions.
2. The called endpoint selects one or more of the media types offered by the calling endpoint for the send and receive directions and returns its selections in a fastStart element in any H.225.0 message up to and including connect. At this point, the called endpoint must be prepared to receive media along any of the channels it selected.
3. If H.245 procedures are needed and one or both of the endpoints do not support tunneling, a separate H.245 connection is used.

Fast Connect is not explicitly configurable. All H.323 Version 2 VoIP endpoints are capable of initiating or accepting fast connect calls. It is assumed that the gateway is capable of sending and receiving fast connect procedures unless its corresponding dial peer has been configured for the Resource Reservation Protocol (RSVP). (In other words, the req-qos is set to a value other than the default of best-effort.) If the dial peer has been configured for RSVP, traditional "slow" connect procedures are followed, and the endpoint neither attempts to initiate fast connect nor responds to a fast connect request from its peer.

A terminating endpoint can reject fast connect by simply omitting the fastStart element from all H.225.0 messages up to and including connect. In this case, normal H.245 procedures are followed and a separate H.245 TCP connection is established. So, if an endpoint does not support the fast connect procedures, normal H.245 procedures are followed. In addition, certain conditions can cause a fast connect call to fall back to normal H.245 procedures to complete the call.

Once a media connection has been opened (an audio path has been established), either endpoint has the option of switching to H.245 procedures (if they are needed) by using H.245 tunneling, whereby H.245 messages are encapsulated within the h245Control element of H.225.0 messages.

The **dtmf-relay** command is the only H.245 cognizant command that can initiate H.245 tunneling procedures from a fast connect call. If H.245 tunneling is active on the call, switching to a separate H.245 connection is not supported.

A Cisco terminating endpoint accepts a fast connect request only if a pair of symmetric codecs (codecs that in both directions are equivalent or identical) can be selected from a list that has been offered. The originating endpoint is constrained only by what it can send through the codec (or voice class codec list) associated with the dial peer.

If the Cisco originating endpoint has offered multiple codecs and the terminating endpoint selects a pair of asymmetric (mismatched) codecs, the originating endpoint initiates separate H.245 procedures to correct the asymmetric codec situation.

Fast connect is backward compatible with H.323 Version 1 configurations.

Note Because fast connect is compliant with H.323 Version 2 and because the majority of endpoints prefer to establish a call by using fast connect procedures, this feature is not configurable. The H.323 fast connect feature does not require any additional configuration beyond a working voice configuration.

H.450.2 Call Transfer

Call transfer allows an H.323 endpoint to redirect an answered call to another H.323 endpoint. Cisco gateways support H.450.2 call transfer as the transferring and transferred-to party. The transferring endpoint must be an H.450-capable terminal; the Cisco gateway cannot act as the transferring endpoint. Gatekeeper-controlled or gatekeeper-initiated call transfer is not supported.

Note Certain devices are limited in their support of H.450. The Cisco 1700 and uBR820 platforms do not support IVR. Therefore, these platforms are not able to act as H.450 transferring endpoints.

H.450.2 specifies two variants of call transfer:

- Transfer without consultation—The transferring endpoint supplies the number of the transferred-to endpoint as part of the transfer request, and the two remote endpoints are transferred together. A Cisco gateway cannot be the transferring endpoint.
- Transfer with consultation—This feature is not currently supported.

H.450.3 Call Deflection

Call deflection is a feature under H.450.3 Call Diversion (Call Forwarding) that allows a called H.323 endpoint to redirect the unanswered call to another H.323 endpoint. Cisco gateways support H.450.3 call deflection as the originating, deflecting, and deflected-to gateway. The Cisco gateway as the deflecting gateway supports invocation of call deflection only by using an incoming PRI QSIG message (call deflection cannot be invoked by using any other trunk type).

If the deflecting endpoint is a Cisco gateway, the telephony endpoint on the PRI of the deflecting gateway invokes call deflection by sending an equivalent QSIG reroute invoke request within a FACILITY message to the gateway. The deflecting gateway then uses the procedures outlined in the H.450.3 call deflection standard to transfer the call to another endpoint. Note that the initiation of deflection using QSIG reroute invoke is valid only on calls that arrived as H.323 calls at the deflecting gateway. In other words, for calls that arrive at the gateway through a telephony interface (such as a hairpin call) or by using a non-H.323 IP protocol, QSIG reroute invoke is ignored.

Cisco H.323 Version 2 software does not support gatekeeper-controlled or gatekeeper-initiated call deflection.

Note Certain devices are limited in their support of the H.450 standard. The Cisco AS5800 universal access server is not able to convert QSIG to H.450. The Cisco 1700 and uBR820 platforms do not support IVR. Therefore, these devices are not able to act as H.450 deflecting endpoints.

Gateway Support for Alternate Endpoints

Alternate endpoints allow a gatekeeper to specify alternative destinations for a call when queried with an ARQ by an originating gateway. If the first destination gateway fails to connect, the gateway tries all the alternate destinations before going to the next dial peer rotary (if a rotary is configured).

Note This feature is not supported by the Cisco gatekeeper; it is intended for use with third-party gatekeepers that implement the alternate endpoint field in the ACF message. No support is provided for the gateway to send a list of alternate endpoints in RRQ messages.

Gatekeeper C Code Generic API for GKTMP in a UNIX Environment

This API allows third-party applications that run in a UNIX host to send GKTMP messages to a Cisco gatekeeper and receive GKTMP messages from a Cisco gatekeeper. This API may be used to develop back-end services such as authentication, billing, and address translation.

Gateway Support for a Network-Based Billing Number

Gateway support for a network-based billing number informs the gatekeeper of the specific voice port or T1/E1 span from which an incoming call entered the ingress gateway. This is done using a Cisco proprietary, nonstandard field that has been added to the ARQ message sent by the ingress gateway. No configuration is necessary for this feature.

Gateway Support for Voice-Port Description

Gateway support for voice-port description provides the gatekeeper with a configurable string that identifies the voice port or T1/E1 span from which an incoming call entered the ingress gateway. This is done using a Cisco proprietary, nonstandard field that has been added to the ARQ message sent by the ingress gateway. The string in the ARQ message corresponds to the setting of the **voice-port description** command.

Gateway support for voice-port description is similar to the network-based billing number feature, but it differs in two important respects:

- The voice-port description field is only included in the ARQ message if the voice-port description is configured through the CLI for the applicable voice port.
- Because the voice-port description is configurable, the user can provide customer-specific information to the gatekeeper. For example, the voice-port description can be configured to correspond to the carrier identification code (CIC) for calls received on a particular T1/E1 span.

H.323 Signaling

When interworking with ISDN, with T-1 channel-associated signaling (CAS), and with E-1 R2 services from the PSTN, H.323 signaling enables VoIP networks to properly signal the setup and teardown of calls. In-band tones and announcements are generated as needed at the originating or terminating switch. When a tone is played at the destination switch, the backward voice path from the called party to the calling party is cut through early so that the calling party can hear the tone or announcement. To prevent fraudulent calls, the voice path is cut through in both directions only after the connect message is received from the destination. The call progress indicator, which signals the availability of in-band communication, is carried end to end as required when interworking with ISDN and CAS protocols.

The H.323 signaling feature prevents unexpected behavior, such as early alerting (when an alert message is returned immediately after a call proceeding message is sent), to ensure that the calling party does not hear conflicting call progress information, such as a ringback tone followed by a busy tone, and does not miss hearing a tone or announcement when one should play. Support for network-side ISDN and reduction in the risk of speech clipping is also addressed.

The H.323 signaling feature is dependent on Cisco H.323 gateways, gatekeepers, and VoIP features.

H.323 signaling provides the following:

- In-Band Tones and Announcements, page 265
- End-to-End Alerting, page 267
- Cut-Through of Voice Path, page 267
- H.245 Initiation, page 267
- Overlap Dialing, page 268

In-Band Tones and Announcements

In-band progress tones and announcements are required for PSTN services and for ISDN speech and 3.1 kHz voice services, per Bellcore and American National Standards Institute (ANSI) specifications. To guarantee that in-band tones and announcements are generated when required and at the appropriate switch, Cisco H.323 signaling software ensures that the progress indicator (PI) is carried end to end in call-signaling messages between the called party and the calling party. The PI in outbound dial peers can also be configured at the H.323 VoIP gateway, if necessary.

The PI is an IE that signals when in-band tones and announcements are available. The PI controls whether the local switch generates the appropriate tone or announcement or whether the remote switch is responsible for the generation. For example, if the terminating switch generates the ringback tone, it sends a PI of 1 or 8 in the alerting message. If the originating switch receives an alerting message without a PI, it generates the ringback tone.

The specific PI that a switch sends in call messages, if any, depends on the model of the switch. To ensure that in-band communication is generated appropriately, it may be necessary in some instances to override the default behavior of the switch by manually configuring the PI at the Cisco H.323 gateway.

The PI is configurable in setup messages from the outbound VoIP dial peer, typically at the originating gateway, and in alert, progress, and connect messages from the outbound POTS dial peer, typically at the terminating gateway. The PI is configured by using the **progress_ind** dial-peer configuration command. Table 22 shows the PI values that may be configured on the H.323 gateway.

Table 22 Configurable Progress Indicator Values for H.323 Gateways

PI	Description	Message Type
0	No progress indicator is included.	Setup
1	Call is not end-to-end ISDN; further call progress information may be available in-band.	Alert, setup, progress, connect
2	Destination address is non-ISDN.	Alert, progress, connect
3	Origination address is non-ISDN.	Setup
8	In-band information or appropriate pattern is now available.	Alert, progress, connect

When the interworking is between ISDN and non-ISDN networks, the originating gateway reacts as follows:

- If the originating switch does not include a PI in setup messages, the originating gateway assumes that the originating switch is ISDN and expects the switch to generate the ringback tone. Determine which device generates the ringback tone by using the **progress_ind** dial-peer configuration command:
  - To enable the terminating switch to generate the ringback tone, set the PI to 8 in the alert messages on the terminating gateway. The progress indicator is configured in the POTS dial peer.
  - To enable the originating gateway to generate the ringback tone, set the PI to 3 in setup messages on the originating gateway. The PI is configured in the VoIP dial peer.

Note If the terminating gateway sends an alert message with no PI value, the originating gateway generates the ringback tone. But if the terminating gateway sends an alert message that has a PI of 1, 2, or 8, the originating gateway does not generate the ringback tone.

- The originating gateway cuts through the voice path in the backward direction when it receives a progress or alert message that has a PI of 1, 2, or 8.

Note Pure ISDN calls may use different protocols at the originating and terminating ends. For example, a call may originate on ETSI and terminate on NI2. If the two protocols are not compatible end to end, the gateway drops all IEs from messages, including the progress indicator. Because a progress indicator is required in all progress messages, the originating gateway inserts a PI of 1 in the progress message. To avoid dropping IEs, use the **isdn gateway-max-internetworking** global configuration command to prevent the gateway from checking protocol compatibility.

End-to-End Alerting

Early alerting is prevented in these ways:

- For calls that terminate at an ISDN switch—The terminating gateway sends an alert message to the originating gateway only after it receives an alert message from the terminating switch.
- For calls that terminate at a CAS switch—The terminating gateway sends a progress message, rather than an alert message, to the originating gateway after it receives a setup message.

Cut-Through of Voice Path

When tones and announcements are generated at the destination switch, the backward voice path from the called party to the calling party is cut through before the tones and announcements are played. This allows announcements, such as "The number you have called has been changed," or allows tones for error conditions, such as network congestion, to be forwarded to the calling party. To prevent fraudulent calls, the originating gateway does not perform full cut-through until it receives a connect message from the destination switch. Cut-through is performed as follows:

- For calls that terminate at an ISDN switch—The terminating gateway performs backward cut-through when it receives an alert or progress message and full cut-through (both directions) when it receives a connect message. The originating gateway performs backward cut-through when it receives a call proceeding message and full cut-through when it receives a connect message.
- For calls that terminate at a CAS switch—The terminating gateway performs backward cut-through after it sends a progress message and full cut-through (both directions) when it receives an off-hook signal. The originating gateway performs backward cut-through when it receives a progress message and full cut-through when it receives a connect message.

Note If the originating or terminating gateway sends a call proceeding message and then receives a call proceeding message with a progress indicator of 1, 2, or 8, the gateway converts this call proceeding message into a progress message with a corresponding PI.

H.245 Initiation

To avoid speech clipping, H.245 capabilities are now initiated at the originating gateway at the earliest possible moment, when the originating gateway receives a call proceeding message from the terminating gateway. Previously, call proceeding messages were not passed end to end across the VoIP network; H.245 was initiated only after the originating gateway received an alert message.

Overlap Dialing

To enhance overlap dialing, the call proceeding message is now passed transparently from the terminating switch to the originating switch if the originating switch does not include the sending complete information element in the setup message. The call proceeding message notifies the originating switch that the terminating switch has collected all dialed digits that are required to route the call. If the originating switch sends a sending complete IE, the originating gateway responds with a call proceeding message, and the session application drops the call proceeding message sent by the terminating switch.

Configurable Timers in H.225.0

When a call is attempted, a TCP connection is made: the TCP socket connection is made for the signaling that the H.225.0 protocol carries. When the timer expires, the call is timed out and attempted using another dial peer, if one has been defined. Cisco configurable timers in H.225.0 software allow users to configure the H.225.0 TCP connection timeout value for all outgoing call attempts (on a per-VoIP dial peer basis).

In previous releases of Cisco IOS software, the call attempt timeout was 15 seconds and could not be changed. In some cases, however, users might need a shorter timeout value to facilitate a faster failover. In other cases, they might need a greater timeout value.

Configurable timers in H.225 address those needs by allowing the user to override the default of 15 seconds and configure the timeout value.

See the “H.323 Configuration Task List” section for information on how to configure timers in H.225.0.

Answer Supervision Reporting

Answer supervision reporting is an enhancement to the information request (IRR) Registration, Admission, and Status (RAS) protocol message that enables gatekeepers to maintain call accounting information by reporting the call connection time of connected calls to the gatekeeper.

In H.323 configurations, the endpoint (gateway) uses direct call-routed signaling. Gatekeepers do not have real-time knowledge or control over the state of a call and are dependent on the endpoints to provide them with necessary real-time information, such as call connect time, call termination time, and call termination reason.

When a call ends, the gateway sends a DRQ message with the BillingInformationToken (which contains the duration of the call) to the gatekeeper. If for some reason the gatekeeper does not receive the DRQ message, the gatekeeper will not have the information about when the call started or the duration of the call, which is necessary to maintain accounting information.

Answer supervision reporting allows the call connection time to be reported to the gatekeeper upon the connection of a call and at periodic intervals thereafter. Answer supervision reporting adds a proprietary Cisco parameter, the call connection time, to the perCallInfo parameter in the nonStandardData field, which is located in the IRR message. When a connect message is received, the originating gateway sends the unsolicited IRR message to its gatekeeper. On sending a connect message, the terminating gateway sends the unsolicited IRR message to its gatekeeper. If the ACF message has a nonzero value for the IRR frequency parameter, the gateway sends the unsolicited IRR message to its gatekeeper at periodic intervals, which are determined by the value in the IRRfrequency parameter.

With the exception of containing the call connection time in the perCallInfo parameter, the IRR message and its functionality remain the same.

Gateway-to-Gatekeeper Billing Redundancy

Gateway-to-gatekeeper billing enhances the accounting capabilities of the Cisco H.323 gateway and provides support for VocalTec™ gatekeepers. Gateway-to-gatekeeper billing redundancy provides for redundant billing information to be sent to an alternate gatekeeper if the primary gatekeeper to which a gateway is registered becomes unavailable.

During the process of establishing a call, the primary gatekeeper sends an ACF message to the registered gateway. The ACF message includes the billing information of the user and an access token. To provide the billing information to an alternate gatekeeper if the primary gatekeeper is unavailable when the call session ends, the access token information sent in the ACF message is also included in the DRQ message that is sent to the alternate gatekeeper.

This features enables the alternate gatekeeper to obtain the billing information required to successfully complete the transaction.

For further information on configuring gateway-to-gatekeeper billing redundancy, refer to *Cisco H.235 Accounting and Security Enhancements for Cisco Gateways, Cisco H.323 Gateway Security and Accounting Enhancements,* and *Gateway Support for Alternate Gatekeeper.*

Ecosystem Gatekeeper Interoperability

Ecosystem gatekeeper interoperability adds support for the alternate gatekeeper field (altGKInfo) in the gatekeeper rejection (GRJ), registration rejection (RRJ), and admission rejection (ARJ) messages. This allows a gateway to move between gatekeepers during the GRQ, RRQ, and ARQ phases. There is no need for gateway reconfiguration or for a gatekeeper failover in the gateway.

Gateways can be configured to switch from their primary gatekeeper to an alternate gatekeeper if a failure or outage occurs. If an outage occurs and gateways move from one gatekeeper to another, there may be an imbalance in the number of gateways registered to each gatekeeper. The ecosystem gatekeeper interoperability feature helps to restore the balance (when the outage has been corrected) by allowing some of the gateways to be moved back to their proper gatekeepers.

The altGKInfo consists of two subfields: the alternateGatekeeper and the altGKisPermanent flag. The alternateGatekeeper is the list of alternate gatekeepers. The altGKisPermanent is a flag that indicates whether the gatekeepers in the associated alternateGatekeeper field are permanent or temporary.

- If the current state of the altGKisPermanent flag is TRUE, the new altGKInfo of any RAS message received from one of the alternate gatekeepers is accepted and the new list will replace the existing list.
- If the current state of the altGKisPermanent flag is FALSE, the altGKInfo of any RAS message received from one of the alternate gatekeepers will be ignored.

If the current permanent gatekeeper becomes nonresponsive and the altGKisPermanent flag is set to FALSE, the gateway sets the internal state of the altGKisPermanent flag to TRUE. This allows the gateway to accept the alternate gatekeeper list from one of the gatekeepers in the existing alternate gatekeeper list.

The handling of the altGKInfo field varies depending on whether it is included in a GRJ or an RRJ message.

For further information on configuring ecosystem gatekeeper interoperability, refer to *Gateway Support for Alternate Gatekeepers, Configuring H.323 VoIP Gateway for Cisco Access Platforms*, and *Ecosystem Gatekeeper Interoperability Enhancements*.

AltGKInfo in GRJ Messages

When the gateway accepts the alternate gatekeeper list from the GRJ, the gateway sends a GRQ message to a gatekeeper on the list. The selection is based on priority of the alternate gatekeepers. Each alternate gatekeeper is tried until a GCF message is received.

If the gateway receives a GRJ message without the AltGKInfo field, it accepts the rejection. Because this is the first phase for the gateway to contact a gatekeeper, the gateway is considered lost without a gatekeeper.

During the GRQ phase, the gateway ignores the value of the altGKisPermanent flag in any RAS message and sets the value internally to TRUE.

AltGKInfo in RRJ Messages

When the gateway accepts the alternate gatekeeper list from the first RRJ message, the gateway retransmits an RRQ message to a gatekeeper on the alternate gatekeeper list. The selection is based on priority of the alternate gatekeepers.

The retransmission of the RRQ message depends on the type of RRQ (full or lightweight), the current state of the altGKisPermanent flag, and the current state of the needToRegister flag of each alternate gatekeeper as follows:

- If the state of the altGKisPermanent flag is TRUE and the state of the needToRegister flag is NO, the gateway will retransmit the full RRQ to an alternate gatekeeper for full RRQs and a lightweight RRQ for lightweight RRQs.
- If the state of the altGKisPermanent flag is TRUE and the state of the needToRegister flag is YES, the gateway will retransmit the full RRQ to an alternate gatekeeper for full RRQs and lightweight RRQs.
- If the state of the altGKisPermanent flag is FALSE and the state of the needToRegister flag is NO, the gateway will retransmit a lightweight RRQ for lightweight RRQs and nothing for full RRQs.
- If the state of the altGKisPermanent flag is TRUE and the state of the needToRegister flag is YES, the gateway will not retransmit the RRQ.

If the gateway receives an RRJ message without the AltGKInfo field, it accepts the rejection and returns to the GRQ phase. If the state of the altGKisPermanent flag is FALSE, the gateway sends the GRQ message to the original gatekeeper that sent the first RRJ. If the state of the altGKisPermanent flag is TRUE, the gateway sends the GRQ to the current gatekeeper.

If the current state of the altGKisPermanent flag is TRUE, then the next RAS message is sent to the new gatekeeper. Otherwise, the next RAS message is sent to the original gatekeeper.

If the gateway exhausts the list of alternate gatekeepers without receiving any response from an alternate gatekeeper, the gateway returns to the GRQ phase.

For more information regarding the Cisco ecosystem gatekeeper interoperability feature, see the “Alternate Gatekeepers” section in the “Configuring H.323 Gateways and Proxies” chapter.

H.323 Restrictions

The following sections contain the restrictions that apply to the Cisco H.323-compliant features:

H.323 Version 2 Feature Restrictions

The following restrictions apply to the Cisco H.323 Version 2 features:

- All systems must be running either Cisco IOS Release 11.3(9)NA and later releases or Cisco IOS Release 12.0(3)T and later releases to interoperate with the Cisco H.323 Version 2 features. Earlier releases contain H.323 Version 1 software that does not support protocol messages that have an H.323 Version 2 protocol identifier. The earlier releases will not interoperate with Cisco H.323 Version 2 Phase 2 features.
- To use H.450 services (call transfer or call deflection), use Cisco IOS Release 12.1(1)T of the gatekeeper: H.450 on the gateways is incompatible with previous releases of the Cisco gatekeeper.
- If a Cisco AS5300 universal access server is used, the software requires the appropriate version of VCWare.
- The H.323 Version 2 fast connect feature is not explicitly configurable. It is assumed that the gateway is capable of sending and receiving fast connect procedures unless its corresponding dial peer has been configured for RSVP (in other words, the req-qos is set to a value other than the default of best-effort). If the dial peer has been configured for RSVP, traditional “slow” connect procedures will be followed, and the endpoint will neither attempt to initiate fast connect nor respond to a fast connect request from its peer.

H.323 Signaling Enhancement Feature Restrictions

The following restrictions apply to the H.323 signaling enhancement feature:

- Supplementary voice services are not supported with ISDN and CAS over an H.323 network—except on the NET5 switch.
- Progress messages require a PI value, and only ITU-T standards are supported.
- Progress indicator 2 is not supported in progress messages for the DMS100 switch.
- TCL 2.0 for IVR supports the interworking signaling enhancements only on the Cisco AS5300. For IVR on other Cisco platforms, select TCL 1.0 as the session application. To use standard IVR applications with TCL 1.0, configure the application name as “session.t.old” by using the **call application voice** global configuration command. It is not necessary to do this if customized scripts are used.

- The Cisco AS5300 universal access server sends a connect message to the originating gateway after it receives a setup message only when it is configured for one of the following supported switch types:
  - 5ESS
  - NET5
  - NTT
  - QSIG
  - QSIGP
- For the SS7 interconnect for voice gateways solution, the following behavior applies to suspend and resume messages, which are supported on NET5 and NI2+ ISDN interfaces:
  - If the ISDN interface is NET5, the Cisco AS5300 sends a notify message with the notification indicator (NI) set to user-suspended or user-resumed.
  - If the ISDN interface is NI2+, the Cisco AS5300 sends a suspend or resume message to the Cisco SC2200.
  - If the Cisco SC2200 receives an ISUP suspend or resume message, it sends an NI2+ suspend or resume message to the Cisco AS5300.
  - Both the Cisco AS5300 and SC2200 timers start when a suspend message is received. The Cisco AS5300 timer, T307, is configurable from 30 to 300 seconds. The Cisco SC2200 timer, T6, is not configurable and has a default of 120 seconds if the ISUP variant Q.761 is used.

    When the Cisco AS5300 and the SC2200 receive a resume message, the timers are stopped. If either of the the timers expires, the call is released with a cause code of normal clearing.

Configurable Timers in H.225.0 Restriction

This feature is limited to H.323 dial peers.

Source Call Signal Address and H.245 Empty Capabilities Set Restrictions

The following restrictions apply to source call signal address and H.245 empty capabilities set:

- To use H.450 services (call transfer or call deflection), Cisco IOS Release 12.1(2)T of the gatekeeper must be used. H.450 on the gateways is incompatible with previous releases of the Cisco gatekeeper.
- If a Cisco AS5300 universal access server is used, the system requires the appropriate version of VCWare.

Ecosystem Gatekeeper Interoperability Restrictions

The following restrictions apply to ecosystem gatekeeper interoperability:

- The maximum number of alternate gatekeepers is eight (including static gatekeepers).
- During the retransmission of the GRQ or RRQ messages, the gateway responds only to the current gatekeeper (regardless of the state of the altGKisPermanent flag).
- The process of retransmission to an alternate gatekeeper can be time-consuming.

H.323 Prerequisite Tasks

To use the Cisco H.323 signaling enhancements, first do the following:

- Establish a working IP network. For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*.
- Install the appropriate voice network module and voice interface card for the Cisco router. For more information about the physical characteristics of the voice network module or on how to install it, refer to the *Voice Network Module and Voice Interface Card Configuration Note* that came with the voice network module.
- Configure your H.323 gateways, gatekeepers, and proxies. For more information about configuring VoIP for your access platform, see the "Configuring H.323 Gateways," "Configuring H.323 Gatekeepers and Proxies," and "Voice over IP Overview" chapters in this configuration guide.
- To ensure network security, configure a RADIUS authentication, authorization, and accounting (AAA) server.

  In addition to the configuration, make sure that the following information is configured in your CiscoSecure AAA server:

  In the /etc/raddb/clients file, ensure that the following information is provided:

```
#Client Name            Key
#-----------       -------------------
gk215.cisco.com         testing123
```

  Where:

  gk215.cisco.com is resolved to the IP address of the gatekeeper requesting authentication

  In the /etc/raddb/users file, ensure that the following information is provided:

```
taeduk@cisco.com  Password = "thiswouldbethepassword"
User-Service-Type = Framed-User,
Login-Service = Telnet
```

  Where:

  taeduk@cisco.com is the h323-id of the gateway authenticating to gatekeeper gk215.cisco.com.
- Configure an NTP server for your network.

Additional requirements and tasks for the individual features follow:

- The configurable timers in the H.225.0 feature require the Cisco H.323 VoIP Gateway for Cisco Access Platforms feature.
- Answer supervision reporting requires a Cisco H.323 gatekeeper.
- Gateway-to-gatekeeper billing redundancy requires a Cisco H.323 gatekeeper and the Gateway Support for Alternate Gatekeepers feature.
- Ecosystem gatekeeper interoperability requires a Cisco H.323 gatekeeper.
- For H.323 Version 2 features, configure an NTP server for the network.

H.323 Configuration Task List

To configure the H.323 features in this chapter, perform the tasks described in the following sections:

Configuring Timers in H.225.0

To use the configurable timers in H.225.0, first create an H.323 voice class and then specify the timeout value associated with that class. To configure the H.225.0 TCP timeout value, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 4	`Router(config)# voice class h323 number`	Enters voice class mode to create or modify an H.323 voice class. The *number* argument identifies the H.323 voice class. There is no default value.
Step 5	`Router(voice-class)# h225 timeout tcp establish value`	Sets the H.225.0 TCP timeout value for the specified voice class. The *value* argument indicates the timeout value, in seconds. There is no default value.
Step 6	`Router(voice-class)# exit`	Exits voice class mode.

Next, associate the H.323 voice class with each VoIP dial peer that should use the specified timeout. To associate the H.323 voice class with a dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice tag voip`	Enters dial-peer configuration mode and defines a remote VoIP dial peer. The keywords and arguments are as follows: • The *tag* argument is one or more digits identifying the dial peer.Valid entries are from 1 to 2,147,483,647. • The **voip** keyword indicates a VoIP peer using voice encapsulation on the IP network.
Step 2	`Router(config-dial-peer)# voice-class h323 number`	Associates the specified H.323 voice class (and all of its related attributes) with the dial peer. The *number* argument identifies the H.323 voice class. There are no default values.

Verifying the H.225.0 TCP Timeout Value

To verify that the timeout value is defined for a dial peer, enter the **show running-config** command. The output shows the current configuration of the voice class and the dial peer.

```
Router# show running-config

Building configuration...

Current configuration:
!
voice class h323 1
 h225 timeout tcp etablish 10

dial-peer voice 919 voip
 application session
 destination-pattern 919555....
 voice-class codec 1
 voice-class h323 1
 session target ras
```

Configuring H.245 Tunneling of DTMF Relay in Conjunction with Fast Connect

The **dtmf-relay** command configured on the outgoing VoIP dial peer initiates H.245 tunneling procedures from a fast connect call. Note that H.245 tunneling will be activated only if the **dtmf-relay**, **h245-alphanumeric**, or **h245-signal** (but *not* **cisco-rtp**) commands are configured on the VoIP dial peer.

Configuring H.450

A Cisco gateway for H.450 is configured in one of the following ways, depending on what the gateway needs to do:

- By redirecting an unanswered call (call deflection).
- By transferring an answered call to a new DN (call transfer without consultation).

Although there are no new CLI commands for configuring H.450 services, the services are enabled only when a TCL/IVR Session Application is configured. Therefore, to use H.450 services, you must configure a TCL/IVR-based "application" on each applicable incoming dial peer for each Cisco gateway that will be involved in call transfer or call deflection. If no special TCL/IVR behavior is required, you can use the standard TCL/IVR application "session." This is not to be confused with application "SESSION," which is not TCL/IVR-based and does not support H.450 services.

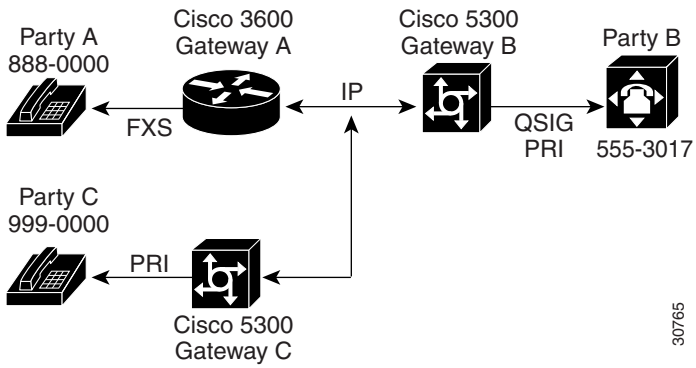
In addition, if call deflection is to be initiated from a QSIG PRI, you must configure the PRI using the **isdn switch-type primary-qsig** and **isdn alert end-to-end** commands.

Note For general information on configuring dial peer application and the meaning of incoming dial peer, refer to *Voice over IP for the Cisco AS5300*.

Configuring Call Deflection

A sample call deflection configuration is shown in Figure 70.

Figure 70 H.450 Configuration to Redirect Unanswered Calls



In this example, three gateways are configured to redirect unanswered calls, so that when Party A calls Party B, Party B can invoke deflection to pass the call to Party C. For this to work, "application session" or another TCL/IVR-based application must be configured on each applicable incoming dial peer as follows:

- On Gateway A, the POTS dial peer for destination pattern 8880000.
- On Gateway B, the VoIP dial peer for destination pattern 8880000.
- On Gateway C, the VoIP dial peer for destination pattern 8880000.

To configure the Gateway A POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *tag* `pots`	Enters dial-peer configuration mode. The *tag* argument is a digit that defines a particular dial peer. Valid entries are from 1 to 2,147,483,647.
Step 2	`Router(dial-peer)# application` *name*	Specifies the application that will be invoked for this dial peer. Only TCL-based applications are able to support H.450 services. The *name* argument indicates the name of the predefined TCL/IVR application. Incoming calls using this POTS dial peer will be handed off to this application.

Command	Purpose
Step 3 `Router(dial-peer)# destination-pattern [+]string[T]`	Specifies either the prefix or the full E.164 telephone number (depending on the dial plan) to be used for a dial peer. The keywords and arguments are as follows: • [+]—(Optional) Specifies a character indicating an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Specifies a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – The asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. Only on the Cisco 3600 router, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,)—Inserts a pause between digits. – Period (.)—Matches any entered digit (this character is used as a wildcard). On the Cisco 3600 router, the period cannot be used as a leading character in a string (for example, .650). – Percent sign (%)—Indicates that the previous digit/pattern occurred zero or multiple times, similar to the wildcard usage in the regular expression. – Plus sign (+)—Matches a sequence of one or more matches of the character/pattern. **Note** The plus sign used as part of the digit string is different from the plus sign that can be used in front of the digit string to indicate that the string is an E.164 standard number. – Circumflex (^)—Indicates a match to the beginning of the string. – Dollar sign ($)—Matches the null string at the end of the input string. – Backslash symbol (\)—Is followed by a single character matching that character or used with a single character having no other significance (matching that character). – Question mark (?)— Indicates that the previous digit occurred zero or one time. – Brackets ([])—Indicates a range of digits. A range is a sequence of characters enclosed in the brackets, and only numeric characters from “0” to “9” are allowed in the range. This is similar to a regular expression rule.

	Command	Purpose
		– Parentheses (())—Indicate a pattern and is the same as the regular expression rule—for example, 408(555). Parentheses are used in conjunction with symbols ?, %, or +. For more information on applying wildcard symbols to destination patterns and the dial strings that result, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter. • **T**—(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 4	`Router(dial-peer)# exit`	Exits dial-peer configuration mode.
Step 5	**2600 and 3600 Series Routers** `Router(config)# port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Specifies the voice slot number, subunit number, and port through which incoming VoIP calls will be received. The keywords and arguments are as follows: • *slot-number*—Specifies the slot number in the Cisco router in which the voice interface card is installed. Valid entries are from 0 to 3, depending on the slot in which it has been installed. • *subunit-number*—Specifies the subunit on the voice interface card in which the voice port is located. Valid entries are 0 or 1. • *port*—Specifies the voice interface card location. Valid entries are 0 or 3. • *slot*—Specifies the router location in which the voice port adapter is installed. Valid entries are from 0 to 3. • *port*—Specifies the voice interface card location. Valid entries are 0 or 3. • *ds0-group-no*—Indicates the defined DS0 group number. Each defined DS0 group number is represented on a separate voice port. This allows individual DS0s to be defined on the digital T1/E1 card. **Note** The slashes must be entered along with the arguments shown within the braces in the Command column.

To configure the VoIP dial peers on Gateways B and C, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number voip`	Enters dial-peer configuration mode. The *number* argument defines a particular dial peer. Valid entries are 1 to 2,147,483,647.
Step 2	`Router(dial-peer)# application name`	Specifies the application that will be invoked for this dial peer. Only Tool Command Language- (TCL-) based applications are able to support H.450 services. The *name* argument indicates the name of the predefined TCL/IVR application. Incoming calls using this VoIP dial peer will be handed off to this application.
Step 3	`Router(dial-peer)# destination-pattern [+]string[T]`	Specifies either the prefix or the full E.164 telephone number (depending on the dial plan) to be used for a dial peer. For a description of the keywords and arguments for this command, see Step 3 in the first configuration task table (showing how to configure the Gateway A POTS dial peer) in the "Configuring Call Deflection" section on page 276.

	Command	Purpose
Step 4	**2600 and 3600 Series VoIP Dial Peers** `Router(dial-peer)#session target {ipv4:destination-address | dns:[$s$. | $d$. | $e$. | $u$.]host-name | loopback:rtp | loopback:compressed | loopback:uncompressed}`	Specifies the network-specific address for a specified dial peer. The keywords and arguments are as follows: • **ipv4:***destination-address*—Specifies the IP address of the dial peer. • **dns:***host-name*—Indicates that the Domain Name System (DNS) will be used to resolve the name of the IP address. Valid entries for this parameter are characters that represent the name of the host device. One of the following four optional wildcards can be used with this keyword when defining the session target for VoIP peers: – **s**—Indicates that the source destination pattern will be used as part of the domain name. – **d**—Indicates that the destination number will be used as part of the domain name. – **e**—Indicates that the digits in the called number will be reversed, that periods will be added in between each digit of the called number, and that this string will be used as part of the domain name. – **u**—Indicates that the unmatched portion of the destination pattern (such as a defined extension number) will be used as part of the domain name.
		• loopback:rtp—Indicates that all voice data will be looped back to the originating source. This is applicable for VoIP peers. • loopback:compressed—Indicates that all voice data will be looped back in compressed mode to the originating source. This is applicable for POTS peers. – **loopback:uncompressed**—Indicates that all voice data will be looped back in uncompressed mode to the originating source. This is applicable for POTS peers.

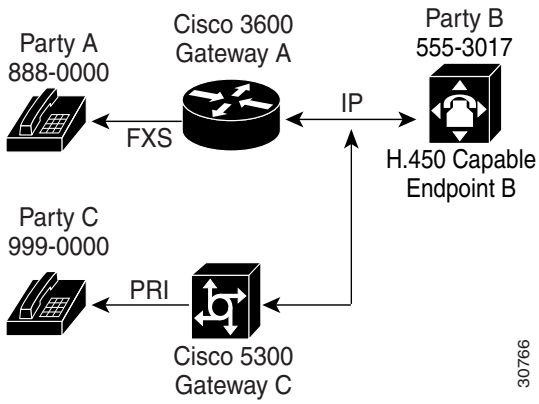
To configure the Gateway B PRI, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**Cisco 4000 Series Access Servers** `Router(config)# interface serial number:timeslot`	Configures the serial interface. The keywords and arguments are as follows: • *number*—Channelized E1 or T1 controller number. • *time-slot*—For ISDN, the D channel time slot, which is the :23 channel for channelized T1 and the :15 channel for channelized E1. PRI time slots are in the range of 0 to 23 for channelized T1 and in the range of 0 to 30 for channelized E1. – For channel-associated signaling or robbed-bit signaling, *time-slot* is the channel group number. – The colon (:) is required. – On a dual port card, it is possible to run channelized on one port and PRI on the other port.
Step 2	`Router(config-if)# isdn switch-type switch-type`	Configures the ISDN interface as a primary QSIG interface. The *switch-type* argument is the service provider switch type. PRI switch types vary by geographic area. (Refer to the command reference master index, or search online for this information.)

Configuring Call Transfer Without Consultation

A sample configuration is shown in Figure 71.

Figure 71 H.450 Configuration for Calls Transfer Without Consultation



In this example, two gateways are configured to handle call transfers without consultation, so that when Party A calls Party B at 555-3017 at Endpoint B, Endpoint B answers and then invokes call transfer to Party C. To do this, configure the application session or another TCL/IVR-based application on each applicable incoming dial peer as follows:

- On Gateway A, the POTS dial peer for destination pattern 8880000.
- On Gateway C, the VoIP dial peer for destination pattern 8880000.

To configure the Gateway A POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *tag* `pots`	Enters dial-peer configuration mode. The *tag* argument is a digit that defines a particular dial peer. Valid entries are from 1 to 2,147,483,647.
Step 2	`Router(dial-peer)# application` *name*	Specifies the application that will be invoked for this dial peer. Only Tool Command Language- (TCL-) based applications are able to support H.450 services. The *name* argument indicates the name of the predefined TCL/IVR application. Incoming calls using this POTS dial peer will be handed off to this application.
Step 3	`Router(dial-peer)# destination-pattern [+]`*string*`[T]`	Specifies either the prefix or the full E.164 telephone number to be used for a dial peer (depending on the dial plan) . For a description of the keywords and arguments for this command, see Step 3 in the first configuration task table (showing how to configure the Gateway A POTS dial peer) in the "Configuring Call Deflection" section on page 276.

	Command	Purpose
Step 4	**2600 and 3600 Series POTS Dial Peers** `Router(dial-peer)# session target`	Specifies the IP address of the destination gateway for "outbound" dial peers. Because this is an "incoming" dial peer, the session target is not applicable, so the IP address is ignored.
Step 5	`Router (dial-peer)# port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Specifies the voice slot number, subunit number, and port through which incoming VoIP calls will be received. For a description of the keywords and arguments for this command, see Step 5 in the first configuration task table (showing how to configure the Gateway A POTS dial peer) in the "Configuring Call Deflection" section on page 276.
Step 6	`Router(dial-peer)# exit`	Exits dial-peer configuration mode.

To configure the Gateway C VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice tag voip`	Enters dial-peer configuration mode. The *tag* argument is a digit that defines a particular dial peer. Valid entries are from 1 to 2,147,483,647.
Step 2	`Router(dial-peer)# application name`	Specifies the application that will be invoked for this dial peer. Only Tool Command Language- (TCL-) based applications are able to support H.450 services. The *name* argument indicates the name of the predefined TCL/IVR application. Incoming calls using this VoIP dial peer will be handed off to this application.
Step 3	`Router(dial-peer)# destination-pattern [+]string[T]`	Specifies either the prefix or the full E.164 telephone number to be used for a dial peer (depending on the dial plan) . For a description of the keywords and arguments for this command, see Step 3 in the first configuration task table (showing how to configure the Gateway A POTS dial peer) in the "Configuring Call Deflection" section on page 276.
Step 4	**2600 and 3600 Series VoIP Dial Peers** `Router(dial-peer)# session target {ipv4:destination address | dns: [$s$. | $d$. | $e$.| $u$.]host-name | loopback:rtp | loopback:compressed | loopback:uncompressed}`	Specifies the network-specific address for a specified dial peer. For a description of the keywords and arguments for this command, see Step 4 in the second configuration task table (showing how to configure the VoIP dial peers on Gateways B and C) in the "Configuring Call Deflection" section on page 276.

For more information about POTS dial peers, refer to the Cisco IOS Release 12.0 *Voice, Video, and Home Applications Configuration Guide* or see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide.

For more information about any of the commands used to configure VoIP dial peers, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*; the *Cisco IOS Voice, Video, and Home Applications Command Reference*; or see the "Configuring Voice Ports" or the "Configuring Voice over IP" chapters in this configuration guide.

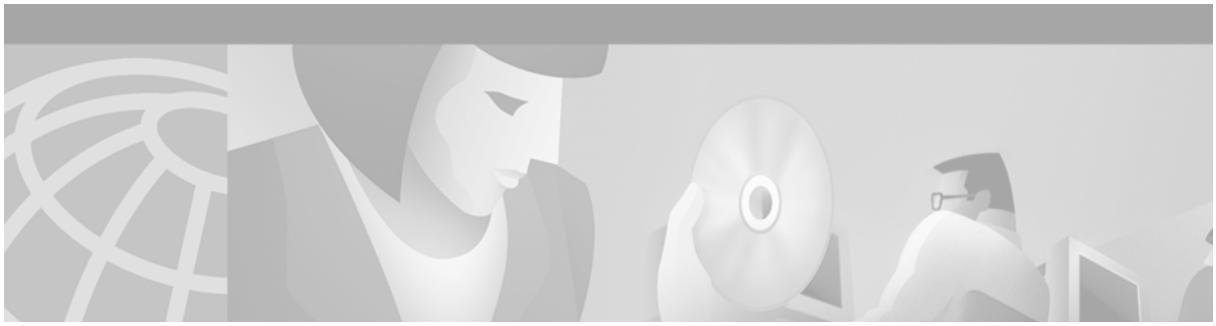
Configuring Voice-Port Descriptions

The voice-port description feature uses the existing **description** subcommand for the voice port. When the voice-port description is being configured, the exact contents of the description field are included in the ARQ message sent from the ingress gateway.

Note Configuring the voice-port description has no effect for calls that are not configured to use RAS.

To configure the description on a voice port, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**2600 and 3600 Series Routers** `Router(config)# voice-port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Enters voice-port configuration mode for the specified voice port. The arguments are as follows: • *slot-number*—Specifies the slot number in the Cisco router in which the voice interface card (VIC) is installed. Valid entries are from 0 to 3, depending on the slot in which it has been installed. • *subunit-number*—Specifies the subunit on the VIC in which the voice port is located. Valid entries are 0 or 1. • *port*—Specifies the voice port number. Valid entries are 0 or 1. • *slot*—Specifies the router location in which the voice port adapter is installed. Valid entries are from 0 to 3. • *port*—Indicates the voice interface card location. Valid entries are 0 or 3. • *dso-group-no*—Indicates the defines DS0 group number. Each defined DS0 group number is represented on a separate voice port. This allows you to define individual DS0s on the digital T1/E1 card.
Step 2	`Router(config-voiceport)# description string`	Defines the description associated with the voice port. The *string* argument is a character string from 1 to 80 characters.



Configuring H.323 Gateways

This chapter describes the configuration of H.323 gateways and contains the following sections:

For a complete description of the gateway commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index or search online. For general information about H.323 gateways and their functions, see the "H.323 Applications" chapter in this configuration guide.

For more information on configuring Cisco mobile telephony products, see Appendix F, "Global System for Mobile Communications Full Rate and Enhanced Full Rate Codecs."

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms"section in the "Using Cisco IOS Software" chapter.

H.323 Gateway Prerequisite Tasks

Before configuring the router as a gateway, perform the following tasks:

- Establish a working IP network. For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*.
- Develop a network plan that details the requirements and characteristics of your Voice over IP (VoIP) network. For further information, see the "Voice over IP Overview" chapter of this configuration guide and refer to the *Voice over IP Implementation Guide*.
- Ensure that the routers you intend to configure as H.323 gateways are running a Cisco IOS software image that contains gateway functionality. (Software images that support gateway features contain *-gw-* in the code image name.)

To use the H.323 security and accounting features described in this document, keep the following in mind:

- These features use the H.235 standard. Because the standard is broad, ensure that the gatekeeper provides H.235 functionality that specifically complements the gateway implementation described in this document.
- In addition, because the H.323 gateway sends the accounting information using a non-standard field in the ClearToken message, ensure that the gatekeeper is able to handle this information.

For more information about specific gatekeepers that can be used with these H.323 security and accounting features, refer to http://von.cisco.com/interoperability/.

H.323 Gateway Configuration Task List

An H.323 gateway is an endpoint on a LAN that provides real-time, two-way communication between H.323 terminals on the LAN and other International Telecommunication Union Telecommunication Standardization Sector (ITU-T) terminals in the WAN. An H.323 gateway can also communicate with another H.323 gateway. Gateways allow H.323 terminals to communicate with non-H.323 terminals by converting protocols. The gateway is the point at which a circuit-switched call is encoded and repackaged into IP packets. Because gateways function as H.323 endpoints, they provide admission control, address lookup and translation, and accounting services. In an environment in which both gatekeepers and gateways are used, only gateways are configured to send VoIP data.

To configure an H.323 gateway, perform the tasks described in the following sections. Except for the first task, all tasks are optional.

Identifying a Router Interface As an H.323 Gateway

To configure a Cisco device as an H.323 gateway in a service provider environment, configure at least one of its interfaces as a gateway interface. Use either an interface that is connected to the gatekeeper or a loopback interface for the gateway interface. The interface that is connected to the gatekeeper is usually a LAN interface—Fast Ethernet, Ethernet, FDDI, or Token Ring.

To configure a gateway interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gateway`	Enables the gateway and enters gateway configuration mode.
Step 2	`Router(config-gateway)# exit`	Exits gateway configuration mode.
Step 3	`Router(config)# ip cef`	(Optional) Enables Cisco Express Forwarding (CEF) routing.

	Command	Purpose
Step 4	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode for the interface that is connected to the gatekeeper. The keywords and arguments are as follow: • *type*—Specifies the type of interface to be configured. • *number*—Specifies the port, connector, or interface card number. The number is assigned at the factory at the time of installation or when added to a system and can be displayed with the **show interfaces** command. • *name-tag*—(Optional) Specifies the logic name to identify the server configuration so that multiple entries of server configuration can be entered.
Step 5	`Router(config-if)# h323-gateway voip interface`	Identifies this interface as a VoIP gateway interface.
Step 6	`Router(config-if)# h323-gateway voip id gatekeeper-id {ipaddr ip-address [port-number] | multicast} [priority number]`	(Optional) Defines the name and location of the gatekeeper for this gateway. The keywords and arguments are as follows: • *gatekeeper-id*—Indicates the H.323 identification of the gatekeeper. This value must exactly match the gatekeeper ID in the gatekeeper configuration. The recommended format is name.domain-name. • **ipaddr**—Indicates that the gateway will use an IP address to locate the gatekeeper. • *ip-address*—Defines the IP address to be used to identify the gatekeeper. • *port-number*— (Optional) Defines the port number used. • **multicast**—Indicates that the gateway will use multicast to locate the gatekeeper. • **priority** *number*—(Optional) The priority of this gatekeeper. The range is 1 through 127, and the default value is 127.
Step 7	`Router(config-if)# h323-gateway voip h323-id interface-id`	(Optional) Defines the H.323 name of the gateway, identifying this gateway to its associated gatekeeper. The *interface-id* argument is the H.323 name (ID) used by this gateway when this gateway communicates with its associated gatekeeper. Usually, this ID is the name of the gateway, with the gatekeeper domain name appended to the end: name@domain-name.
Step 8	`Router(config-if) h323-gateway voip tech-prefix prefix`	(Optional) Defines the technology prefix that the gateway will register with the gatekeeper. The *prefix* argument defines the numbers used as the technology prefixes. Each technology prefix can contain up to 11 characters. Although not required, a pound symbol (#) is frequently used as the last digit in a technology prefix. Valid characters are 0 through 9, the pound symbol (#), and the asterisk (*).

Verifying Gateway Interface Configuration

To find the current registration information and status of the gateway, use the **show gateway** command.

Configuring Gateway RAS

The Registration, Admission, and Status (RAS) signaling function performs registration, admissions, status, and disengage procedures between the H.323 VoIP gateway and the H.323 VoIP gatekeeper. RAS tells the gatekeeper to translate the E.164 phone number of the session target into an IP address.

In the RAS exchange between a gateway and a gatekeeper, a technology prefix is used to identify the specific gateway when the selected zone contains multiple gateways. The **tech-prefix** dial-peer configuration command is used to define technology prefixes. See the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide for more information on the **tech-prefix** command, or refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

In most cases there is a dynamic protocol exchange between the gateway and the gatekeeper that enables the gateway to inform the gatekeeper about technology prefixes and where to forward calls. If, for some reason, that dynamic registry feature is not in effect, statically configure the gatekeeper to query the gateway for this information. To configure the gatekeeper to query for this information, see the "Configuring H.323 Gatekeepers and Proxies" chapter. To configure RAS, define specific parameters for the applicable plain old telephone service (POTS) and VoIP dial peers. The POTS dial peer informs the system of which voice port to direct incoming VoIP calls to and (optionally) determines that RAS-initiated calls will have a technology prefix prepended to the destination telephone number. The VoIP dial peer determines how to direct calls that originate from a local voice port into the VoIP cloud to the session target. The session target indicates the address of the remote gateway where the call is terminated. There are several different ways to define the destination gateway address:

- By statically configuring the IP address of the gateway.
- By defining the Domain Name System (DNS) of the gateway.
- By using RAS. If RAS is being used, the gateway determines the destination target by querying the RAS gatekeeper.

To configure RAS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode to configure a POTS peer. The *number* argument is a tag that identifies the dial peer. (This number has local significance only.) Valid entries are from 1 to 2,147,483,647.
Step 2	`Router(config-dial-peer)# destination-pattern [+]`*string*`[T]`	Specifies the E.164 address associated with this dial peer. The keywords and arguments are as follows: • **+**—(Optional) Specifies a character indicating an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Indicates a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – The asterisk (*) and pound sign (#)—Indicates the keys that appear on standard touch-tone dial pads. On the Cisco 3600 series routers only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,)—Inserts a pause between digits. – Period (.)—Matches any entered digit (this character is used as a wildcard). On the Cisco 3600 series routers, the period cannot be used as a leading character in a string (for example, .650). – Percent sign (%)—Indicates that the previous digit/pattern occurred zero or multiple times, similar to the wildcard usage in the regular expression. – Plus sign (+)—Matches a sequence of one or more matches of the character/pattern. **Note** The plus sign used as part of the digit string is different from the plus sign that can be used in front of the digit string to indicate that the string is an E.164 standard number. – Circumflex (^)—Indicates a match to the beginning of the string. – Dollar sign ($)—Matches the null string at the end of the input string. – Backslash symbol (\)—Is followed by a single character matching that character or used with a single character having no other significance (matching that character). – Question mark (?)— Indicates that the previous digit occurred zero or one time.

	Command	Purpose
		– Brackets ([])—Indicate a range of digits. A range is a sequence of characters enclosed in the brackets, and only numeric characters from "0" to "9" are allowed in the range. This is similar to a regular expression rule. – Parentheses (())—Indicate a pattern and is the same as the regular expression rule—for example, 408(555). Parentheses are used in conjunction with symbols ?, %, or +. For more information on applying wildcard symbols to destination patterns and the dial strings that result, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter. • **T**—(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.
Step 3	**Cisco AS5300 Universal Access Server** `Router(config-dial-peer)# port controller:D`	Associates this POTS dial peer with a specific voice port. The keywords and arguments are as follows: • *controller*—Specifies the T1 or E1 controller. • **:D**—Indicates the D channel associated with the ISDN PRI. **Note** The syntax of the **port** command is platform specific. For information on how to configure this command for your specific device, see the **port** command documentation in the "Configuring Voice Ports" chapter.
Step 4	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode.
Step 5	`Router(config)# dial-peer voice tag voip`	Enters dial-peer configuration mode to configure a VoIP peer. The *tag* argument identifies the dial peer. (This number has local significance only.) Valid entries are from 1 to 2,147,483,647.
Step 6	`Router(config-dial-peer)# destination-pattern [+]string[T]`	For an explanation of the command, keywords, and arguments, see Step 2 of this configuration task table.
Step 7	`Router (config-dial-peer)# tech-prefix number`	The *number* argument defines the numbers used as the technology prefix. Each technology prefix can contain up to 11 characters. Although not strictly necessary, a pound symbol (#) is frequently used as the last digit in a technology prefix. Valid characters are 0 though 9, the pound symbol (#), and the asterisk (*).
Step 8	`Router (config-dial-peer)# session target ras`	Specifies that the RAS protocol is being used to determine the IP address of the session target—meaning that a gatekeeper will translate the E.164 address to an IP address.

Verifying RAS Configuration

To verify the POTS and VoIP dial-peer configuration, use the **show dial-peer voice** command. The following example shows output for a VoIP dial peer using RAS on a Cisco AS5300 universal access server:

```
Router# show dial-peer voice 1234

VoiceOverIpPeer1234
tag = 1234, destination-pattern = 1234',
answer-address = ',
group = 1234, Admin state is up, Operation state is up,
incoming called-number = ', connections/maximum = 0/unlimited,
application associated:
type = voip, session-target = ras',
technology prefix: 8#
ip precedence = 0, UDP checksum = disabled,
session-protocol = cisco, req-qos = controlled-load,
acc-qos = best-effort,
fax-rate = voice, codec = g729r8,
Expect factor = 10, Icpif = 30,
VAD = enabled, Poor QOV Trap = disabled,
```

Troubleshooting Tips

To troubleshoot the dial-peer configuration, perform the following tasks:

- To display the types and addressing of RAS messages sent and received, use the **debug ras** command. The debug output lists the message type using mnemonics defined in ITU-T specification H.225.
- To display additional information about the actual contents of the H.225 RAS messages, use the **debug h225 asn1** command.

Configuring AAA Functionality on the Gateway

For the gateway to provide authentication and accounting services, enable and configure your gateway to support authentication, authorization, and accounting (AAA) services. AAA enables the gateway to interact with a RADIUS security server to authenticate users (typically incoming calls) and to perform accounting services. For more information about RADIUS and AAA security services, refer to the *Cisco IOS Security Configuration Guide*.

AAA Authentication

The gateway normally uses AAA in conjunction with interactive voice response (IVR) to check the legitimacy of a prospective gateway user on the basis of an account number (collected by IVR) or Automatic Number Identification (ANI). When the gateway uses AAA with IVR, the IVR application collects the user account and personal identification number (PIN) information and then passes it to the AAA interface. The AAA interface makes a RADIUS authentication request using the given information and, based on the information received from the RADIUS server, forwards either a pass message or a fail message to the IVR application.

For more information about configuring IVR, see the “Configuring Interactive Voice Response” chapter. For more information about authentication services using AAA, refer to the “Configuring Authentication” chapter in the *Cisco IOS Security Configuration Guide*.

AAA Accounting

A call leg is a discrete segment of a call connection that lies between two points in the connection. Each call made through the gateway consists of two call legs: incoming and outgoing. The RADIUS server collects basic start-stop connection accounting data or syslog accounting information during the accounting process for each call leg created on the gateway.

To collect basic start-stop connection accounting data, the gateway must be configured to support gateway-specific H.323 accounting functionality. The gateway sends accounting data to the RADIUS server in one of four ways, as is shown in the following sections:

- Using RADIUS AV Pairs, page 292
- Appendix , "Using RADIUS AV Pairs"Overloading the Acct-Session-Id Field, page 293
- Using Vendor-Specific RADIUS Attributes, page 294
- Using Syslog Records, page 295

Using RADIUS AV Pairs

Basic start-stop connection accounting data and standard RADIUS attributes are used where possible using standard Internet Engineering Task Force (IETF) RADIUS attribute/value (AV) pairs. Table 23 shows the IETF RADIUS attributes that are supported.

Table 23 Supported IETF RADIUS Accounting Attributes

Number	Attribute	Description
30	Called-Station-Id	Allows the network access server to send the called telephone number as part of the Access-Request packet (using Dialed Number Identification Service [DNIS] or similar technology). This attribute is only supported on ISDN and on modem calls on the Cisco AS5200 and Cisco AS5300 routers if used with ISDN PRI.
31	Calling-Station-Id	Allows the network access server to send the calling telephone number as part of the Access-Request packet (using ANI or similar technology). This attribute has the same value as the remote-addr attribute from TACACS+. This attribute is supported only on ISDN and on modem calls on the Cisco AS5200 and Cisco AS5300 routers if used with ISDN PRI.
42	Acct-Input-Octets	Indicates how many octets have been received from the port over the course of the accounting service being provided.
43	Acct-Output-Octets	Indicates how many octets have been sent to the port over the course of delivering the accounting service.
44	Acct-Session-Id	Indicates a unique accounting identifier that makes it easy to match start and stop records in a log file. Acct-Session-Id numbers restart at 1 each time the router is power-cycled or the software is reloaded.
47	Acct-Input-Packets	Indicates how many packets have been received from the port over the course of this service being provided to a framed user.
48	Acct-Output-Packets	Indicates how many packets have been sent to the port in the course of delivering this service to a framed user.

For more information about RADIUS and the use of IETF-defined attributes, refer to the *Cisco IOS Security Configuration Guide*.

Overloading the Acct-Session-Id Field

Attributes that cannot be mapped to standard RADIUS attributes are packed into the Acct-Session-Id attribute field as ASCII strings separated by the "/" character. The Acct-Session-Id attribute contains the RADIUS account session ID, which is a unique identifier that links accounting records associated with the same login session for a user. To support additional fields, the following string format has been defined for this field:

```
<session id>/<call leg setup time>/<gateway id>/<connection id>/<call origin>/
<call type>/<connect time>/<disconnect time>/<disconnect cause>/<remote ip address>
```

Table 24 shows the field attributes to be used with the Overloaded Acct-Session-Id method and provides a brief description of each.

Table 24 Field Attributes in Overloaded Acct-Session-Id

Field Attribute	Description
SESSION-ID	Specifies the standard RADIUS account session ID.
SETUP-TIME	Provides the Q.931 setup time for this connection in Network Time Protocol (NTP) format. NTP time formats are displayed as %H:%M:%S.%k %Z %tw %tn %td %Y where: • %H is hour (00 to 23). • %M is minutes (00 to 59). • %S is seconds (00 to 59). • %k is milliseconds (000 to 999). • %Z is time zone string. • %tw is day of week (Saturday through Sunday). • %tn is month name (January through December). • %td is day of month (01 to 31). • %Y is year including century (for example, 1998).
GATEWAY-ID	Indicates the name of the underlying gateway in the form of "gateway.domain_name."
CALL-ORIGIN	Indicates the origin of the call relative to the gateway. Possible values are **originate** and **answer**.
CALL-TYPE	Indicates call leg type. Possible values are **telephony** and **VoIP**.
CONNECTION-ID	Specifies the unique global identifier used to correlate call legs that belong to the same end-to-end call. The field consists of 4 long words (128 bits). Each long word is displayed as a hexadecimal value and is separated by a space character.
CONNECT-TIME	Provides the Q.931 connect time for this call leg, in NTP format.
DISCONNECT-TIME	Provides the Q.931 disconnect time for this call leg, in NTP format.

Table 24 Field Attributes in Overloaded Acct-Session-Id

Field Attribute	Description
DISCONNECT-CAUSE	Specifies the reason a call was taken offline as defined in the Q.931 specification.
REMOTE-IP-ADDRESS	Indicates the address of the remote gateway port where the call is connected.

Because of the limited size of the Acct-Session-Id string, it is not possible to embed many information elements in it. Therefore, this feature supports only a limited set of accounting information elements.

Use the **gw-accounting h323** command to configure the overloaded session ID method of applying H.323 gateway-specific accounting.

Using Vendor-Specific RADIUS Attributes

The IETF draft standard specifies a method for communicating vendor-specific information between the network access server (NAS) and the RADIUS server by using the vendor-specific attribute (Attribute 26). Vendor-specific attributes (VSAs) allow vendors to support their own extended attributes that are not suitable for general use. The Cisco RADIUS implementation supports one vendor-specific option using the format recommended in the specification. The Cisco vendor-ID is 9, and the supported option has vendor-type 1, which is named "cisco-avpair." The value is a string of the format:

```
protocol: attribute sep value *
```

"Protocol" is a value of the Cisco "protocol" attribute for a particular type of authorization. "Attribute" and "value" are an appropriate AV pair defined in the Cisco TACACS+ specification, and "sep" is "=" for mandatory attributes and "*" for optional attributes. The full set of features available for TACACS+ authorization can also be used for RADIUS.

For further information on vendor-specific RADIUS attributes, refer to the *RADIUS Vendor-Specific Attributes Voice Implementation Guide* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/vsaig3.htm

The VSA fields and their ASCII values are listed in Table 25.

Table 25 VSA Fields and Their ASCII Values

IETF RADIUS Attribute	Vendor-Specific Company Code	Subtype Number	Attribute Name	Description
26	9	23	h323-remote-address	Indicates the IP address of the remote gateway.
26	9	24	h323-conf-id	Identifies the conference ID.
26	9	25	h323-setup-time	Indicates the setup time for this connection in Coordinated Universal Time (UTC), formerly known as Greenwich Mean Time (GMT) and Zulu time.

Table 25 VSA Fields and Their ASCII Values (continued)

IETF RADIUS Attribute	Vendor-Specific Company Code	Subtype Number	Attribute Name	Description
26	9	26	h323-call-origin	Indicates the origin of the call relative to the gateway. Possible values are **originating** and **terminating,** which are equivalent to **originate** and **answer** in the Call-Origin field.
26	9	27	h323-call-type	Indicates call leg type. Possible values are **telephony** and **VoIP**.
26	9	28	h323-connect-time	Indicates the connection time for this call leg in UTC.
26	9	29	h323-disconnect-time	Indicates the time this call leg was disconnected in UTC.
26	9	30	h323-disconnect-cause	Specifies the reason a connection was taken offline per the Q.931 specification.
26	9	31	h323-voice-quality	Specifies the impairment/calculated planning impairment factor (ICPIF) affecting voice quality for a call.
26	9	33	h323-gw-id	Indicates the name of the underlying gateway.

Use the **gw-accounting h323 vsa** command to configure the VSA method of applying H.323 gateway-specific accounting.

Using Syslog Records

The syslog accounting option exports the information elements associated with each call leg through a system log message, which can be captured by a syslog daemon on the network. The syslog output consists of the following:

```
<server timestamp> <gateway id> <message number> : <message label> : <list of AV pairs>
```

The syslog message fields are listed in Table 26.

Table 26 Syslog Message Output Fields

Field	Description
server timestamp	The time stamp created by the server when it receives the message to log.
gateway id	The name of the gateway that emits the message.
message number	The number assigned to the message by the gateway.
message label	A string that identifies the message category.
list of AV pairs	A string consisting of <attribute name> <attribute value> pairs separated by commas.

Use the **gw-accounting h323 syslog** command to configure the syslog record method of gathering H.323 accounting data.

To configure RADIUS authentication and accounting services (as facilitated through authentication, authorization, and accounting [AAA]), use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) security services.
Step 2	`Router(config)# gw-accounting {h323 [vsa] | syslog}`	Configures gateway-specific H.323 accounting, which may be **h323** or **syslog**. The keywords are as follows: • **h323**—Enables standard H.323 accounting using standard IETF RADIUS attributes. • **vsa**—(Optional) Enables H.323 accounting using RADIUS vendor-specific attributes. • **syslog**—Enables the system logging facility to output accounting information in the form of a system message. **Note** Because the Acct-Session-Id attribute is a standard IETF RADIUS attribute, use the **gw-accounting h323** command to gather accounting data using the overloaded Acct-Session-Id attribute.
Step 3	`Router(config)# aaa authentication login h323 radius`	Sets AAA authentication at login. The keywords are as follows: • **h323**—Defines a method list called h323. • **radius**—Specifies that the RADIUS security protocol be used.
Step 4	`Router(config)# aaa accounting connection h323 start-stop radius`	Defines a method list called **h323**. The keywords are as follows: • **start-stop**—Sends a start accounting notice at the beginning of a process and a stop accounting notice at the end of a process. The start accounting record is sent in the background. The requested user process begins regardless of whether the start accounting notice was received by the accounting server. • **radius**—Specifies that only the RADIUS security protocol be used.

	Command	Purpose
Step 5	`Router(config)# radius-server host ip-address auth-port number acct-port number`	Identifies the RADIUS server and the ports that will be used for authentication and accounting services. The keywords and arguments are as follows: • *ip-address*—Specifies the IP address of the RADIUS server host. • **auth-port**—Specifies User Datagram Protocol (UDP) for authentication requests. • *number*—Specifies the port number for authentication requests; the host is not used for authentication if set to 0. The default authentication port number is 1645. • **acct-port**—Specifies the UDP destination port for accounting requests. • *number*—Port number for accounting requests; the host is not used for accounting if set to 0. The default accounting port number is 1646.
Step 6	`Router(config)# radius-server key key`	Specifies the password used between the gateway and the RADIUS server. The *key* argument specifies the authentication and encryption key used between the router and the RADIUS daemon running on this RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used. The *key* is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax. This is because the leading spaces are ignored, but spaces within and at the end of the key are used. If the key includes spaces, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

Verifying AAA and RADIUS Configuration

To view the configured RADIUS and AAA parameters for this gateway, use the **show running-config** command.

Configuring H.235 Gateway Security

The Cisco H.235-based security and accounting features provide an alternative means for securing H.323 calls. Before Cisco IOS Release 12.0(7)T, only RAS and AAA were used to configure the security and accounting functions for H.323 calls. The H.235-based security and accounting features described in this section can be used by a gatekeeper, which is considered a known and trusted entity, to authenticate, authorize, and route H.323 calls.

The Cisco H.323 gateway supports the use of CryptoH323Tokens for authentication. The CryptoH323Token is defined in the ITU-T H.225 Version 2 standard and is used in a "password-with-hashing" security scheme as described in section 10.3.3 of the H.235 specification.

A cryptoToken can be included in any RAS message to authenticate the sender of the message. A separate database can be used for user ID and password verification.

Cisco H.323 gateways support three levels of authentication:

- Endpoint—The RAS channel used for gateway-to-gatekeeper signaling is not a secure channel. To ensure secure communication, H.235 allows gateways to include an authentication key in their RAS messages. This key is used by the gatekeeper to authenticate the source of the messages. At the endpoint level, validation is performed on all messages from the gateway. The cryptoTokens are validated using the password configured for the gateway.

  Note To secure the RAS messages and calls, it is essential that the gatekeeper provides authentication based on the secure key. The gatekeeper must support H.235 security using the same security scheme as the Cisco gateway.

- Per-Call—When the gateway receives a call over the telephony leg, it prompts the user for an account number and PIN. These two numbers are included in certain RAS messages sent from the endpoint to authenticate the originator of the call.
- All—This option is a combination of the other two. With this option, the validation of cryptoTokens in admission request (ARQ) messages is based on an the account number and PIN of the user who is making a call. The validation of cryptoTokens sent in all the other RAS messages is based on the password configured for the gateway.

CryptoTokens for registration requests (RRQs), unregistration requests (URQs), disengage requests (DRQs), and the terminating side of ARQs contain information about the gateway that generated the token. The cryptoTokens include the gateway identification (ID)—which is the H.323 ID configured on the gateway—and the gateway password. The cryptoTokens for the originating-side ARQ messages contain information about the user that is placing the call, including the user ID and PIN.

Although the scenarios in this document describe how to use the security and accounting features in a prepaid call environment, these features may also be used to authorize IP calls that originate in another domain (inter-service provider or inter-company calls).

The H.235-based security and accounting features can be used in conjunction with AAA. The gateway can be configured to use the gatekeeper for call authentication or authorization, and AAA can be used for call accounting.

In addition, the H.235-based security and accounting features include support for the following:

- Settlement with the gatekeeper, which allows the gateway to obtain, track, and return accounting information.
- Call metering, which allows the gateway to terminate a call if it exceeds the allotted time (in the case of prepaid calls).

Note The H.235 security and accounting features described in this document are separate from, and should not be confused with, the standard interactive voice response (IVR) and AAA features used to authenticate inbound calls or with the settlement functions provided by the Open Settlement Protocol (OSP).

Settlement with the Gatekeeper

The H.235 security and accounting features are designed to support a variety of situations in which some form of authentication or tracking is required. The security features allow control access through the use of a userID-password database. The accounting enhancements allow call usage to be tracked at the origin and at the destination.

Fields have been added to the RAS messages to enhance the accounting capabilities of the Cisco H.323 gateway. These fields allow the gateway to report call-usage information to the gatekeeper. The call-usage information is included in the DRQ message that is sent when the call is terminated.

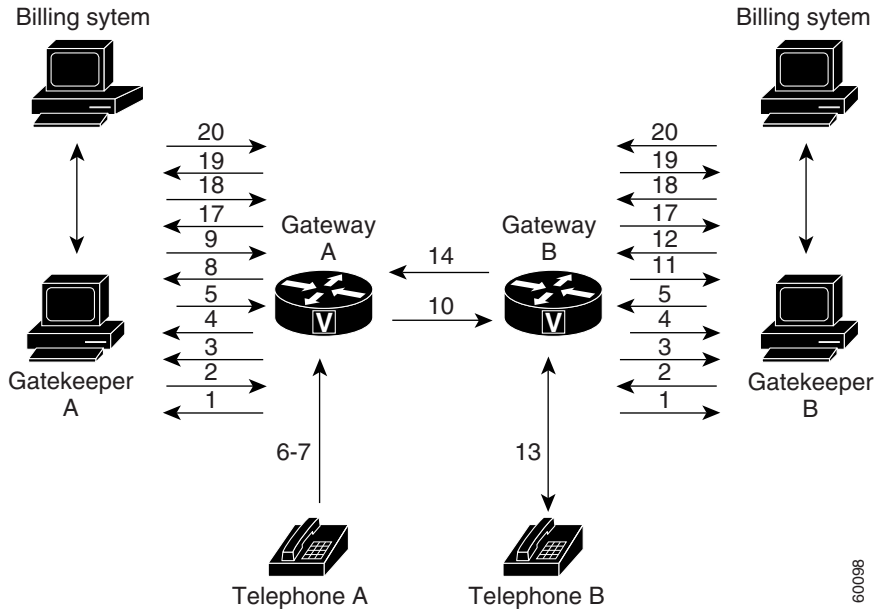
Call Tracking

With prepaid calling services, an account number and PIN must be entered and the duration of the call must be tracked against the remaining credit of the customer. The Cisco H.323 gateway monitors prepaid account balances and terminates a call if the account is exceeded.

Note Because the authentication information includes a time stamp, it is important that all the Cisco H.323 gateways and the gatekeepers (or other entity that is performing the authentication) be synchronized. The Cisco H.323 gateways must be synchronized using the Network Time Protocol (NTP). illustrates the flow of a possible call for which H.323 security and accounting features are used.Flow for a Call That Requires H.323 Security and Accounting Features.

Figure 72 illustrates the flow of a possible call for which H.323 security and accounting features are used.

Figure 72 Flow for a Call That Requires H.323 Security and Accounting Features



In this example, Telephone A is attempting to establish a phone call to Telephone B. The following numbered explanations correspond to the action taking place at each numbered reference in Figure 1.

Gateways Establish Secure Communication with the Gatekeepers

1. Gateways A and B send gatekeeper request (GRQ) messages to their respective gatekeepers. The GRQ message includes the authentication capability and the algorithm object ID.
2. Gatekeepers A and B respond to their respective gateways with gatekeeper confirmation (GCF) messages. The GCF message includes the authentication capability and the algorithm object ID.
3. If the values for the H.323 security parameters do not match what is expected, the gatekeeper responds with a gatekeeper rejection (GRJ) message that contains a reject reason of securityDenial. This prompts the gateway to resend the GRQ.
4. Gateways A and B send registration request (RRQ) messages to their respective gatekeepers. The RRQ message includes authentication information in the cryptoToken field.
5. Gatekeepers A and B respond to their respective gateways with registration confirmation (RCF) messages.

   If an authentication failure occurs, the gatekeeper responds with a registration rejection (RRJ) message.

Secure Telephone Communications Are Initiated

6. Telephone A establishes a connection with Gateway A.
7. Gateway A initiates the interactive voice response (IVR) script to obtain the account number and PIN of the user as well as the desired destination telephone number.
8. Gateway A sends an admission request (ARQ) message to Gatekeeper A. The gateway must include additional information in the ARQ message to enable the gatekeeper to authenticate the call. The information included in the ARQ message varies depending on whether the ARQ message is being sent by the source or the destination gateway. At this point in the scenario, it is the source gateway that is requesting admission. Therefore, the ARQ message includes the account number and PIN of the user. This information is encrypted using MD5 hashing and is included in the cryptoTokens field.
9. Gatekeeper A validates the authentication information, resolves the destination telephone number, and determines the appropriate destination gateway (which is Gateway B in this case). Then Gatekeeper A sends an admission confirmation (ACF) message to Gateway A. The ACF message includes the billing information of the user (such as a reference ID and current account balance for prepaid call services) and an access token.
10. Gateway A sends a setup message to Gateway B. The setup message also includes the access token.
11. Gateway B sends an ARQ message to Gatekeeper B. The ARQ message includes the access token received from Gateway A.
12. Gatekeeper B validates the authentication information in the access token and responds to Gateway B with an ACF message.

    If the authentication information is in error, Gatekeeper B sends an admission rejection (ARJ) message to Gateway B with a reject reason of securityDenial.
13. Gateway B initiates a call to the destination telephone.
14. When the destination telephone is answered, Gateway B sends a connect message to Gateway A.
15. Gateways A and B start their timers to meter the call. If the caller is using prepaid call services, the meter is constantly compared to the account balance of the user, which was included in the ACF message sent in Step 9.

Telephone Communications Are Terminated

16. The call is terminated when one of the parties hangs up or, in the case of prepaid call services, when either of the gateways determines that the account balance of the user has been exceeded.

17. Gateways A and B send DRQ messages to the their respective gatekeepers. The DRQ message contains the resulting billing information.
18. Gatekeepers A and B send disengage confirmation (DCF) messages to their respective gateways.

Communication Between the Gateways and the Gatekeepers Is Terminated

19. Gateways A and B send URQ messages to their respective gatekeepers.
20. Gatekeepers A and B send unregistration confirmation (UCF) messages to their respective gateways.

Downloading IVR Scripts

The Tool Command Language (TCL) IVR scripts are the default scripts for all Cisco voice features that use IVR.

The H.323 security and accounting enhancements described in this document require the use of one of the following IVR scripts:

- voip_auth_acct_pin_dest.tcl
- voip_auth_acct_pin_dest_2.tcl

Note For more information on TCL IVR applications, see the "Configuring TCL IVR Applications" chapter.

voip_auth_acct_pin_dest.tcl

The voip_auth_acct_pin_dest.tcl script does the following:

- Prompts the caller to enter an account number, PIN, and destination number. This information is provided to an H.323 gatekeeper, which authenticates and authorizes the call.

  If the caller is using a debit card account number, the following will occur:

  - The gatekeeper returns the remaining credit time amount.
  - The TCL script monitors the time remaining and, based on a configured value, plays a "time running out" message to the caller. The message (such as, "You have only 3 minutes remaining on your credit.") is played only to the calling party. The called party hears silence during this time. For example, if the configured time-out value is 3 minutes, the message is played when the caller has only 3 minutes of credit left.
  - The TCL script plays a warning message when the credit of the user has been exhausted. The message (such as, "Sorry, you have run out of credit.") is played only to the calling party. The called party hears silence during this time.

- Allows the caller to make subsequent calls to different destinations without disconnecting from the call leg. Thus, the caller is required to enter the account ID and PIN only once (during initial authorization). For making subsequent calls, the caller needs to enter only the destination number. After completing a call to one destination, the caller can disconnect the call by pressing the pound (#) key on the keypad and holding it down from 1 to 2 seconds. If the # key is pressed down for more than 1 second, it is treated as a long pound (#). The called party is disconnected, and the caller is prompted to enter a new destination number. Once a new destination number is entered, the call is authenticated and authorized using this number and the previously provided account number and PIN.

  This feature also allows the caller to continue making additional calls if the called party hangs up.

- Reauthenticates and authorizes each new call. Each time a caller enters a new destination number, the TCL script reauthenticates or authorizes the call with the gatekeeper and, if the caller is using a debit card account, obtains the remaining credit time information.
- Allows the caller to enter the necessary information without having to hear all or any of the prompts. The TCL script will stop playing (or will not begin playing) the prompt if it detects that the caller wants to enter the information without listening to the prompt.

Note The normal terminating character for the account number, PIN, and destination number is the pound (#) key.

- Allows the caller to interrupt announcements by pressing the touch tone key. This TCL script stops playing announcements when the system detects that the caller has pressed any touch tone key.
- Allows the caller to interrupt partially entered numbers and restart from the beginning by pressing a designated key on the keypad. The asterisk (*) key is configured as the interrupt key in the TCL script. The caller can use the asterisk key to cancel an entry and then reenter the account number, PIN, or destination number. The caller is allowed to re-enter a field only a certain number of times. The number of retries may be configured. The default is three times.
- Can terminate a field by size instead of the terminating character (#). The TCL script allows a specified number of digits to be entered in the account number and PIN fields. This means that the caller can type all the digits (without the terminating character) and the script determines how to extract different fields from the number strings. If the caller uses the terminating character, the terminating character takes precedence and the fields are extracted accordingly.
- Supports two languages. The IVR script supports two languages, which must be similar in syntax. The languages must be similar in the manner in which numbers are constructed—especially for currency, amount, and time. All the prompts are recorded and stored in both languages. The language selection is made when the caller presses a predefined key in response to a prompt (such as, “For English, press 1. For Spanish, press 2.”). The TCL script uses the selected language until the caller disconnects.

voip_auth_acct_pin_dest_2.tcl

The voip_auth_acct_pin_dest_2.tcl script is a simplified version of the voip_auth_acct_pin_dest.tcl script. It prompts the caller for an account number followed by a PIN. The caller is then prompted for a destination number. This information is provided to the H.323 gatekeeper that authenticates and authorizes the call. This script provides prompts only in English.

If the caller is using a debit account number, it plays a “time running out” message when the caller has 10 seconds of credit time remaining. It also plays a “time has expired” message when the credit of the caller has been exhausted.

H.235 Gateway Security Configuration Tasks

To use the H.235 security features for routing H.323 calls as illustrated above, do the following:

- Enable H.323 security on the gateway.
- Download the appropriate TCL IVR scripts from the Cisco Connection Online Software Support Center. The URL to this site is as follows:

  http://www.cisco.com/cgi-bin/tablebuild.pl/tclware
- Configure the IVR inbound dial peer on the gateway router.

To enable security on the gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gateway`	Enters gateway configuration mode.
Step 2	`Router(config-gateway)# security password` *password* `level {endpoint | per-call | all}`	Enables security and specifies the level of validation to be performed. The keywords and arguments are as follows: • *password*—Specifies the gateway password. • **endpoint**—Specifies that validation be performed on all RAS messages sent by the gateway using the cryptoTokens that are generated based on the security password configured for the gateway. • **per-call**—Specifies that validation be performed only on the admission messages from the H.323 endpoints to the gateway ARQ messages). The gateway prompts the user for an account number and PIN. These two numbers are sent from the endpoint and are used to authenticate the originator of the call. • **all**—This option is a combination of the **endpoint** and **per-call** options. Specifies that validation be performed on all RAS messages sent by the gateway. The validation of cryptoTokens in ARQ messages is based on the account number and PIN of the user making the call, and the validation of cryptoTokens sent in all other RAS messages is based on the password configured for the gateway.
Step 3	`Router(config-gateway)# exit`	Exits gateway configuration mode.
Step 4	`Router(config)# dial-peer voice` *tag* `pots`	Enters the dial-peer configuration mode to configure a POTS dial peer. The *tag* value of the dial-peer voice POTS command uniquely identifies the dial peer. Valid entries are from 1 to 2,147,483,647.
Step 5	`Router(config-dial-peer)# call application voice` *application-name location* {*word*}	Enters the command to initiate the IVR application and the selected TCL application name. Enter the application name and the location where the TCL IVR script is stored. The arguments are as follows: • *application-name*—Specifies the character string that defines the name of the application. • *location*—Specifies the location of the TCL file in URL format. Valid storage locations are TFTP, FTP, and Flash. • *word*—Specifies the text string that defines an attribute-value (AV) pair specified by the TCL script and understood by the RADIUS server.
Step 6	`Router(config-dial-peer)# destination-pattern [+]`*string*`[T]`	Specifies the E.164 address associated with this dial peer. For an explanation of the keywords and arguments, see Step 2 of the configuration table in the "Configuring Gateway RAS" section on page 288.

	Command	Purpose
Step 7	**Cisco AS5300 Universal Access Server** `Router(config-dial-peer)# port` *controller number*`:D`	Configures the voice port associated with this dial peer. • *controller number*—Specifies the T1 or E1 controller. • **:D**—Indicates the D channel associated with the ISDN PRI. **Note** The syntax of the **port** command is specific to Cisco hardware platforms. For information on how to configure this command for a specific device, refer to the **port** command documentation in the *Cisco IOS Voice, Video, and Fax Command Reference*.

Verifying H.235 Gateway Security Configuration

To display the security password and level when it is enabled, use the **show running-config** command. By default, security is disabled.

```
Router# show running-config
security password 151E0A0E level all
```

Configuring Alternate Gatekeeper Support

An alternate gatekeeper provides redundancy for a gateway in a system in which gatekeepers are used. A gateway may use up to two alternate gatekeepers as a backup in the case of a primary gatekeeper failure.

A gatekeeper manages H.323 endpoints in a consistent manner, allowing them to register with the gatekeeper and to locate another gatekeeper. The gatekeeper provides logic variables for proxies or gateways in a call path to provide connectivity with the Public Switched Telephone Network (PSTN), to improve quality of service (QoS), and to enforce security policies. Multiple gatekeepers may be configured to communicate with one another, either by integrating their addressing into the DNS or by using Cisco IOS configuration options.

Alternate gatekeeper support has the following restrictions:

- This feature can be used only with a gatekeeper that supports the alternate gatekeeper functionality.
- The timer/retry number of RAS messages remains internal to the gateway as currently implemented. This feature does not include commands to allow tuning of these parameters.
- The alternate gatekeeper list is volatile—when the gateway loses power or is reset or reloaded, the alternate gatekeeper list that has been acquired from the gatekeeper is lost.

Gatekeeper Clustering

With gatekeeper clustering there is the potential that bandwidth may be overcommitted in a cluster. For example, suppose that there are five gatekeepers in a cluster and that they share 10 Mbps of bandwidth. Suppose that the endpoints registered to those alternates start placing calls quickly. It is possible that within a few seconds, each gatekeeper could be allocating 3 Mbps of bandwidth if the endpoints on each of the gatekeepers request that much bandwidth. The net result is that the bandwidth consumed in the cluster is 15 Mbps.

The alternate gatekeeper was purposely designed to restrict bandwidth because there is no clear way to sync bandwidth information quickly and efficiently. To work around this problem, "announcement" messages were restricted to intervals as small as 10 seconds. If the gatekeepers get into a situation in which endpoints request bandwidth rapidly, the problem will be discovered and corrective action will take place within at least 10 seconds. Assuming that the gatekeepers are probably not all synchronized on their timers, the announcement messages from the various gatekeepers are likely to be heard more quickly. Therefore, the problem will be less severe. The potential exists, however, for overcommitment of the bandwidth between announcement messages if the call volume increases substantially in a short amount of time (as small as 10 seconds).

Note If you monitor your bandwidth, it is recommended that you consider lowering the maximum bandwidth so that if "spikes" such as those described above do occur, some bandwidth will still be available.

To configure alternate gatekeeper support on a gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface Ethernet 0/1`	Enters interface configuration mode for the selected Ethernet interface.
Step 2	`Router(config-if)# ip address`	Identifies the IP address of the Ethernet interface.
Step 3	`Router(config-if)# h323-gateway voip interface`	Identifies this interface as a Voice over IP (VoIP) gateway interface.
Step 4	`Router(config-if)# h323-gateway voip id` *gatekeeper-id* {**ipaddr** *ip-address* [*port-number*]| **multicast**} [**priority** *number*]	Identifies the gatekeeper for this gateway interface and sets the attributes. For an explanation of the keywords and arguments, see Step 6 in the "Identifying a Router Interface As an H.323 Gateway" section on page 286.
Step 5	`Router(config-if)# h323-gateway voip id` *gatekeeper-id* {**ipaddr** *ip-address* [*port-number*] | **multicast**} [**priority** *number*]	To identify the alternate gatekeeper, use the following keywords and arguments: • *gatekeeper-id*—Indicates the H.323 identification of the gatekeeper. This value must exactly match the gatekeeper identification (ID) in the gatekeeper configuration. The recommended format is *name.domain-name*. • **ipaddr**—Indicates that the gateway will use an IP address to locate the gatekeeper. • *ip-address*—Defines the IP address used to identify the gatekeeper. • *port-number*—(Optional) Defines the port number used. • **multicast**—Indicates that the gateway will use multicast to locate the gatekeeper. • **priority** *number*—(Optional) Specifies the priority of this gatekeeper. The range is 1 through 127, and the default value is 127.

	Command	Purpose
Step 6	`Router(config-if)# h323-gateway voip h323-id` *interface-id*	Identifies the H.323 ID of a particular H.323 endpoint, which in this case is the gateway. The *interface-id* argument is the H.323 name (ID) used by this gateway when this gateway communicates with its associated gatekeeper. Usually, this ID is the name of the gateway, with the gatekeeper domain name appended to the end: name@domain-name.

Verifying Configuration of the Alternate Gatekeeper

To see that there is an alternate gatekeeper configured, enter the **show gate** command

```
Alternate Gatekeeper List
    priority 126 id GK1 ipaddr 172.18.193.61 1719
    priority 127 id GK2 ipaddr 172.18.193.63 1719
```

Configuring Dual Tone Multifrequency Relay

Dual tone multifrequency (DTMF) is the tone generated on a touch-tone phone when the keypad digits are pressed. During a call, DTMF may be entered to access interactive voice response (IVR) systems, such as voice mail and automated banking services.

Although DTMF is usually transported accurately when using high-bit-rate voice codecs such as G.711, low-bit-rate codecs such as G.729 and G.723.1 are highly optimized for voice patterns and tend to distort DTMF tones. As a result, IVR systems may not correctly recognize the tones.

DTMF relay solves the problem of DTMF distortion by transporting DTMF tones “out of band,” or separate from the encoded voice stream.

For a more thorough explanation of DTMF relay, see the “H.323 Applications” chapter.

To configure DTMF relay on a gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *tag* `voip`	Defines a Voice over IP (VoIP) dial peer and enters dial-peer configuration mode. The keywords and argument are as follows: • *tag*—Indicates the digit that defines a particular dial peer. Valid entries are from 1 to 2,147,483,647. • **voip**—Indicates that this is a VoIP peer using voice encapsulation on the POTS network. Use this keyword to configure DTMF relay.

	Command	Purpose
Step 2	`Router(config-dial-peer)# dtmf-relay [cisco-rtp] [h245-alphanumeric] [h245-signal]`	Forwards DTMF tones. The keywords are as follows: • **cisco-rtp**—(Optional) Forwards DTMF tones by using RTP with a Cisco proprietary payload type. • **h245-alphanumeric**—(Optional) Forwards DTMF tones by using the H.245 "alphanumeric" User Input Indication (UII) method. It supports tones 0 through 9, *, #, and A through D. Use this keyword to configure DTMF relay. • **h245-signal**—(Optional) Forwards DTMF tones by using the H.245 "signal" UII method. It supports tones 0 through 9, *, #, and A through D.
Step 3	`Router(config-dial-peer)# codec` *codec* {**clear-channel** | **g711alaw** | **g711ulaw** | **g723ar53** | **g723ar63** | **g723r53** | **g723r63** | **g726r16** | **g726r24** |**g726r32** | **g726r53** | **g726r63** |**g728** | **g729abr8** | **g729ar8** | **g729br8** | **g729r8** | **gsmefr** | **gsmfr**} [**bytes** *payload_size*]	Specifies the voice coder rate of speech for a dial peer. • The *codec* keywords are as follows: – **clear-channel**—Clear channel at 64,000 bits per second (bps). – **g711ala**—G.711 a-Law at 64,000 bits per second. – **g711ula**—G.711 u-Law at 64,000 bps. – **g723ar53**—G.723.1 Annex A at 5300 bps. – **g723ar63**—G.723.1 Annex A at 6300 bps. – **g723r53**—G.723.1 at 5300 bps. – **g723r63**—G.723.1 at 6300 bps. – **g726r16**—G.726 at 16,000 bps. – **g726r24**—G.726 at 24,000 bps. – **g726r32**—G.726 at 32,000 bps. – **g728**—G.728 at 16,000 bps. – **g729abr8**—G.729 Annex A and B at 8000 bps. – **g729ar8**—G729 Annex A at 8000 bps. – **g729br8**—G.729 Annex B at 8000 bps. – **g729r8**—G.729 at 8000 bps. This is the default codec. – **gsmefr**—Global System for Mobile Communications Enhanced Full Rate (GSMEFR) at 12,200 bps. – **gsmfr**—Global System for Mobile Communications Full Rate (GSMFR) at 13,200 bps.

	Command	Purpose
		• **bytes**—(Optional) Specifies the number of bytes in the voice payload of each frame. • *payload-size*—(Optional) The number of bytes in the voice payload of each frame. Refer to the **codec** (dial-peer) command table titled "Voice Payload-Per-Frame Options and Defaults" in the *Cisco IOS Voice, Video, and Fax Command Reference* for valid entries and default values.
Step 4	`Router(config-dial-peer)# destination-pattern [+]string[T]`	Specifies the prefix, the full E.164 telephone number, or an ISDN directory number to be used for a dial peer (depending on the dial plan). For an explanation of the keywords and arguments, see Step 2 of the configuration task table in the "Configuring Gateway RAS" section on page 288.

Step 5	**Cisco 2600 and 3600 Series Routers**
	`Router(config-dial-peer)# session target {ipv4:destination-address | dns:[$s$. | $d$. | $e$. | $u$.] host-name | loopback:rtp | loopback:compressed | loopback:uncompressed}`
	Cisco AS5300 Universal Access Server
	`Route(config-dial-peer)# session target {ipv4:destination-address | dns:[$s$. | $d$. | $e$. | $u$.] host-name | loopback:rtp | loopback:compressed | loopback:uncompressed | mailto:{name | $d$.}@domain-name}`
		Specifies a network-specific address for a specified dial peer or destination gatekeeper.
		Keywords and arguments are as follows:
		Cisco 2600 and 3600 Series Routers
		• **ipv4:***destination-address*—IP address of the dial peer.
		• **dns:***host-name*—Indicates that the DNS will be used to resolve the name of the IP address. Valid entries for this parameter are characters representing the name of the host device. (Optional) You can use one of the following four wildcards with this keyword when defining the session target for VoIP peers:
		– **s.**—Indicates that the source destination pattern will be used as part of the domain name.
		– **d.**—Indicates that the destination number will be used as part of the domain name.
		– **e.**—Indicates that the destination pattern is used as part of the domain name in reverse dotted format for tpc.int DNS format. For example, if the destination number is 310 555-1234 and the session target is configured as e.cisco.com, the translated DNS name will be 4.3.2.1.5.5.5.0.1.3.cisco.com**.**
		– **u.**—Indicates that the unmatched portion of the destination pattern (such as a defined extension number) will be used as part of the domain name.
		• **loopback:rtp**—Indicates that all voice data will be looped back to the originating source. This is applicable for VoIP peers.

Command	Purpose
	• **loopback:compressed**—Indicates that all voice data will be looped back in compressed mode to the originating source. This is applicable for POTS peers. • **loopback:uncompressed**—Indicates that all voice data will be looped back in uncompressed mode to the originating source. This is applicable for POTS peers. **Cisco AS5300 Universal Access Server** In addition to the above, the following keywords and arguments apply to the Cisco AS5300 universal access server: • **mailto:***name*—Specific recipient e-mail address, name, or mailing list alias. • **@***domain-name*—Specifies the appropriate domain name associated with the e-mail address.

Configuring FXS Hookflash Relay

A "hookflash" indication is a brief on-hook condition during a call. The indication is not long enough in duration to be interpreted as a signal to disconnect the call.

PBXs and telephone switches are frequently programmed to intercept hookflash indications and use them as a way to allow a user to invoke supplemental services. In a traditional telephone network, a hookflash results in a voltage change on the telephone line. Because there is no equivalent of this voltage change in an IP network, a message may be sent over the IP network that represents a hookflash. To send a hookflash indication using this message, an H.323 endpoint sends an H.245 user input indication message that contains an "H.245-signal" or "H.245-alpha" structure.

Hookflash relay is supported on the Cisco 2600, 3600, and 7200 series routers and on the MC3810 multiservice concentrator.

For a further explanation of configuring hookflash relay, see the "H.323 Applications" chapter.

To configure hookflash relay on a gateway, use the following commands beginning in global configuration mode:

Note Hookflash relay is enabled only when the **dtmf-relay h245-signal** command is configured on the applicable VoIP dial peers. Hookflash is relayed using an h245-signal indication and can be sent only when an h245-signal is available.

	Command	Purpose

Step 1	**Cisco 2600 and 3600 Series Routers**
	`Router(config)# voice-port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`
	
	Cisco 7200 Series Routers
	`Router(config)# voice-port {slot/port:ds0-group-no} | {slot-number/subunit-number/port}`
	
	Cisco MC3810 Multiservice Concentrator
	`Router(config)# voice-port slot/port`
		Enters voice-port configuration mode.
		
		The keywords and arguments are as follows:
		
		Cisco 2600 and 3600 Series Routers
		• *slot-number*—Specifies the slot number in the Cisco router in which the voice interface card is installed. Valid entries are from 0 to 3, depending on the slot in which it has been installed.
		• *subunit-numbe*r—Specifies the subunit on the voice interface card in which the voice port is located. Valid entries are 0 or 1.
		• *port*—Indicates the voice port. Valid entries are 0 or 1.
		• *slot*—Specifies the router location in which the voice port adapter is installed. Valid entries are from 0 to 3.
		
		Cisco 7200 Series Routers
		• *slot*—Specifies the router location in which the voice port adapter is installed. Valid entries are from 0 to 3.
		• *port*—Indicates the voice interface card location. Valid entries are 0 or 1.*ds0-group-no*—Defines DS0 group number. Each defined DS0 group number is represented on a separate voice port. This allows you to define individual DS0s on the digital T1/E1 card.
		• *slot-number*—Indicates the slot number in the Cisco router in which the voice interface card is installed. Valid entries are from 0 to 3, depending on the slot in which it has been installed.
		• *subunit-number*—Indicates the subunit on the voice interface card in which the voice port is located. Valid entries are 0 or 1.
		• *port*—Indicates the voice port number. Valid entries are 0 or 1.
		
		Cisco MC3810 Multiservice Concentrator
		• *slot*—Specifies the slot number in the Cisco router in which the voice interface card is installed. The only valid entry is 1.
		• *port*—Specifies the voice port number. Valid ranges are as follows:
		– Analog voice ports: from 1 to 6.
		– Digital T1: from 1 to 24.
		– Digital E1: from 1 to 15 and from 17 to 31.

	Command	Purpose
Step 2	`Router(config-voice-port)# timing hookflash-input duration`	Specifies the maximum duration of a hookflash indication. If the hookflash lasts longer than the specified limit, the Foreign Exchange Station (FXS) interface processes the indication as an on-hook. The *duration* is shown in milliseconds. Possible values are 50 through 1550. The default value is 600 milliseconds.
Step 3	`Router(config-voice-port)# timing hookflash-out duration`	Specifies the duration, in milliseconds, of the hookflash indications that the gateway generates on a Foreign Exchange Office (FXO) interface. Valid entries are from 50 through 1550 milliseconds. The default is 400 milliseconds.

Configuring Multiple Codecs

Normally only one codec is specified when a dial peer is configured on a gateway. However, a prioritized list of codecs can be configured to increase the probability of establishing a connection between endpoints during the H.245 exchange phase. For more information about configuring multiple codecs, see the "Codec Negotiation" section in the "H.323 Applications" chapter.

To configure multiple codecs for a dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class codec tag`	Enters voice class configuration mode and assigns an identification tag number for a codec voice class. The *tag* argument is the unique number assigned to the voice class. The valid range is from 1 to 10,000. Each tag number must be unique on the router.
Step 2	`Router(config-class)# codec preference value codec-type [bytes payload-size]`	Adds codecs to the prioritized list of codecs. The keywords and arguments are as follows: • *value*—Specifies the order of preference, with 1 being the most preferred and 12 being the least preferred. • *codec-type*—Specifies the type of codec preferred. Types are as follows: – **clear-channel**—Clear channel at 64,000 bps. – **g711alaw**—G.711a-Law at 64,000 bps. – **g711ulaw**—G.711 u-Law at 64,000 bps. – **g723ar53**—G.723.1 Annex-A at 5300 bps. – **g723ar63**—G.723.1 Annex-A at 6300 bps. – **g723r53**—G.723.1 at 5300 bps. – **g723r63**—G.723.1 at 6300 bps. – **g726r16**—G.726 at 16,000 bps. – **g726r24**—G.726 at 24,000 bps.

		– **g726r32**—G.726 at 32,000 bps. – **g728**—G.728 at 16,000 bps.**g729abr8**—G.729 Annex-A and B at 8000 bps. – **g729br8**—G.729 Annex-B at 8000 bps. – **g729r8**—G.729 at 8000 bps. This is the default codec. – **gsmefr**—Global System for Mobile Communications Enhanced Full Rate (GSMEFR) at 12,200 bps. – **gsmfr**—Global System for Mobile Communications Full Rate (GSMFR) at 13,200 bps. • **bytes**—(Optional) Specifies that the size of the voice frame is in bytes. – *payload-size*—(Optional) Number of bytes that you specify as the voice payload of each frame. Values depend on the codec type and the packet voice protocol.
Step 3	`Router(config-class)# exit`	Exits voice class configuration mode.
Step 4	`Router(config)# dial-peer voice` *tag* `voip`	Enters dial-peer configuration mode to configure a VoIP peer. The *tag* value of the **dial-peer voice** VoIP command uniquely identifies the dial peer. (This number has local significance only.)
Step 5	`Router(config-dial-peer)# voice-class codec` *tag*	The *tag* is the unique number assigned to the voice class. The valid range for this tag is from 1 to 10,000. The tag number maps to the tag number created using the **voice class codec** global configuration command.

Verifying Multiple Codecs Configuration

To show the codecs defined for a particular prioritized list of codecs, use the **show running-config** command.

Configuring Rotary Calling Pattern

Rotary calling pattern routes an incoming call that arrives over a telephony interface back out through another telephony interface under certain circumstances. Rotary calling pattern primarily provides reliable service during network failures. Call establishment using rotary calling pattern is supported by rotary group support of dial peers, where multiple dial peers may match a given destination phone number and be selected in sequence. In addition, if the destinations need to be tried in a certain order, preference may be assigned. Use the **preference** command when configuring the dial peers to reflect the preferred order (0 being the highest preference and 10 the lowest).

If several dial peers match a particular destination pattern, the system attempts to place a call to the dial peer configured with the highest preference. If the call cannot be completed because of a system outage (for example, the gatekeeper or gateway cannot be contacted), the rotary call pattern performs the following tasks:

- Lists all the conditions under which this instance occurs.
- Retries the call to the next highest preference dial peer.
- Continues until no more matching dial peers are found.

If there are equal priority dial peers, the order is determined randomly.

Note The hunting algorithm precedence may be configured. See the **preference** command discussed in the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

Configuring H.323 Support for Virtual Interfaces

H.323 support for virtual interfaces allows the IP address of the gateway to be configured so that the IP address included in the H.323 packet is always the source IP address of the gateway, regardless of the physical interface and protocol used. This single-address feature allows firewall applications to be easily configured to work with H.323 messages.

To configure a source IP address for a gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type slot/port*	Enters interface configuration mode to configure parameters for the specified interface. The keywords and arguments are as follows: • *type*—Indicates the type of interface. • *slot*—Specifies the number of the slot being configured. • *port*—Specifies the number of the port being configured. **Note** This syntax will vary, depending on the platform.
Step 2	`Router(config-if)# h323-gateway voip bind srcaddr` *ip-address*	Sets the source IP address to be used for this gateway. The *ip-address* argument specifies the IP address to be used for outgoing H.323 traffic, which includes H.225, H.245, and RAS messages. Typically, this is the IP address assigned to the Ethernet interface.

Note The **h323-gateway voip bind srcaddr** command can be used on any interface in the router. Using the command on one interface assigns the source address for the entire router.

Verifying the Source IP Address of the Gateway

To verify the source IP address of the gateway, enter the **show running-config** command. The output shows the source IP address that is bound to the interface.

```
router# show running-config

interface Loopback0
 ip address 10.0.0.0 255.255.255.0
 no ip directed-broadcast
 h323-gateway voip bind srcaddr 10.0.0.0
!
interface Ethernet0/0
 ip address 172.18.194.50 255.255.255.0
 no ip directed-broadcast
 h323-gateway voip interface
 h323-gateway voip id j70f_2600_gk2 ipaddr 172.18.194.53 1719
 h323-gateway voip h323-id j70f_3640_gw1
 h323-gateway voip tech-prefix 3#
.
.
.
```

In the following example, Ethernet interface 0/0 is used as the gateway interface. For convenience, the **h323-gateway voip bind srcaddr** command has been specified on the same interface. The designated source IP address is the same as the IP address assigned to the interface.

```
interface Ethernet0/0
 ip address 172.18.194.50 255.255.255.0
 no ip directed-broadcast
 h323-gateway voip interface
 h323-gateway voip id j70f_2600_gk2 ipaddr 172.18.194.53 1719
 h323-gateway voip h323-id j70f_3640_gw1
 h323-gateway voip tech-prefix 3#
 h323-gateway voip bind srcaddr 172.18.194.50
```

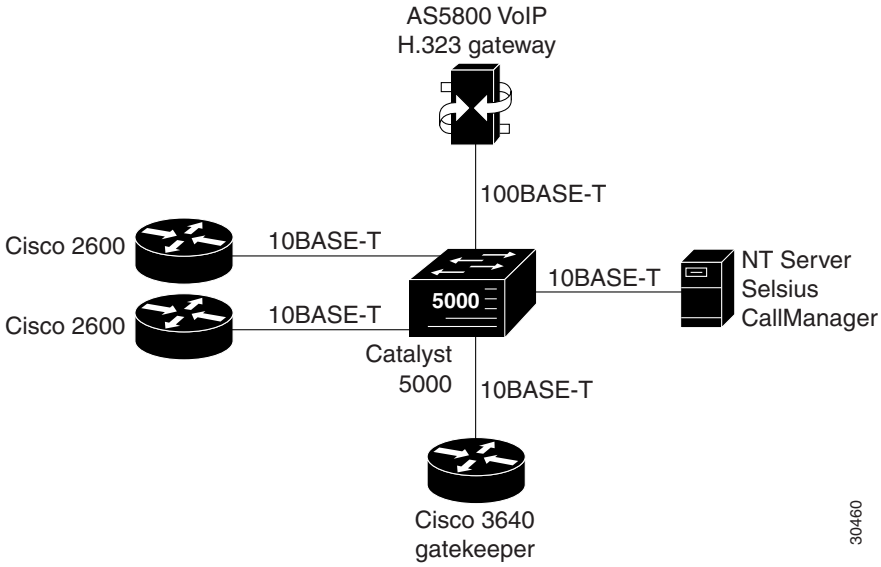
H.323 Gateway Configuration Examples

This section includes the following gateway configuration examples:

H.323 Gateway RAS Configuration Example

Figure 73 shows a Cisco 2600 router and a Cisco AS5800 universal access server as gateways and a Cisco 3640 router as a gatekeeper.

Figure 73 VoIP for the Cisco AS5800



The following example shows a Cisco AS5800 universal access server configured as a gateway using RAS:

```
! Configure the T1 controller. (This configuration is for a T3 card.)
controller T1 1/0/0:1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
! Configure POTS and VoIP dial peers.
dial-peer voice 11111 pots
 incoming called-number 12345
 destination-pattern 9#11111
 direct-inward-dial
 port 1/0/0:1:D
 prefix 11111
!
dial-peer voice 12345 voip
 destination-pattern 12345
 tech-prefix 6#
 session target ras
!
! Enable gateway functionality.
gateway
!
! Enable Cisco Express Forwarding.
ip cef
!
! Configure and enable the gateway interface.
interface FastEthernet0/3/0
 ip address 172.16.0.0.255.255.255.0
 no ip directed-broadcast
 no keepalive
```

```
 full-duplex
 no cdp enable
h323-gateway voip interface
 h323-gateway voip id gk3.gg-dn1 ipaddr 172.18.0.0 1719
 h323-gateway voip h323-id gw3@gg-dn1
 h323-gateway voip tech-prefix 9#
!
! Configure the serial interface.(This configuration is for a T3 serial interface.)
interface Serial1/0/0:1:23
 no ip address
 no ip directed-broadcast
 ip mroute-cache
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no cdp enable
```

AAA Functionality on the Gateway Configuration Example

The following example shows AAA functionality configured on a Cisco AS5300 universal access server.

```
version 12.2
no service single-slot-reload-enable
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
service password-encryption
service tcp-small-servers
!
hostname doc-rtr53-05

aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHENT none
aaa authentication login h323 group radius
aaa authentication ppp default if-needed local
aaa authorization exec default local if-authenticated
aaa authorization exec NO_AUTHOR none
aaa authorization commands 15 default local if-authenticated
aaa authorization commands 15 NO_AUTHOR none
aaa accounting exec default start-stop group tacacs+
aaa accounting exec NO_ACCOUNT none
aaa accounting commands 15 default stop-only group tacacs+
aaa accounting commands 15 NO_ACCOUNT none
aaa accounting connection h323 start-stop group radius
enable secret 5 $1$l545$V4haZZN8AOKem8T1DhF5i/
enable password 7 060506324F41
!
resource-pool disable
!
call rsvp-sync
ip subnet-zero
no ip source-route
no ip finger
ip domain-name the.net
ip name-server 172.22.53.210
ip name-server 172.19.23.12
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
```

```
controller T1 1
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 3
 framing esf
 clock source line secondary 3
 linecode b8zs
 pri-group timeslots 1-24
!
gw-accounting h323
gw-accounting voip
translation-rule 1
 Rule 1 408555.... 5
 Rule 2 650 5
!
interface Loopback0
 ip address 172.21.10.10 255.255.255.255
!
interface Loopback1
 ip address 172.21.104.254 255.255.255.0
!
interface Ethernet0
 no ip address
 shutdown
!
interface Virtual-Template1
 description Template for Multilink Users
 ip unnumbered Loopback1
 no logging event link-status
 no snmp trap link-status
 peer default ip address pool addr-pool
 ppp authentication chap
 ppp multilink
!
interface Serial0:23
 description description Headquarters 324-1937 active PRI line
 no ip address
 no logging event link-status
 no snmp trap link-status
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial1:23
 no ip address
 no logging event link-status
 no snmp trap link-status
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial2:23
 no ip address
 no logging event link-status
 no snmp trap link-status
```

```
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial3:23
 no ip address
 no logging event link-status
 no snmp trap link-status
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 172.21.101.23 255.255.255.0
 duplex auto
 speed auto
!
radius-server host 10.10.10.10 auth-port 1612 acct-port 1616
radius-server retransmit 3
radius-server key 7 00071C080A520E
!
dial-peer voice 1 pots
!
dial-peer voice 2 voip
 destination-pattern +1234...
 session target ipv4:10.1.1.1
!
dial-peer voice 3 voip
 destination-pattern 555*
 session target ipv4:10.1.1.2
!
dial-peer voice 4 voip
 destination-pattern 555
 session target ipv4:10.1.2.2
!
dial-peer voice 90 voip
 destination-pattern 555.%
 session target ipv4:10.1.2.2
!
dial-peer voice 50 voip
 destination-pattern 408555%
 session target ipv4:10.1.1.2
!
dial-peer voice 55 voip
 destination-pattern 408555.%
 session target ipv4:10.2.2.2
!
line con 0
 exec-timeout 0 0
 authorization commands 15 NO_AUTHOR
 authorization exec NO_AUTHOR
 login authentication NO_AUTHENT
 transport input none
line 1 48
 autoselect during-login
 autoselect ppp
 modem InOut
 transport preferred none
 transport output telnet
line aux 0
line vty 0 4
 password 7 03470A1C140635495C
```

```
 transport preferred none
 transport input telnet
 transport output telnet
!
!
!ntp clock-period 17180261
ntp update-calendar
ntp server 172.22.255.1 prefer
```

H.323 Gateway Security Configuration Example

The following example illustrates H.323 security configuration on a Cisco AS5300 gateway.

```
hostname um5300
!
enable password xyz
!
resource-pool disable
!
clock timezone EST -5
clock summer-time EDT recurring
ip subnet-zero
no ip domain-lookup
!
isdn switch-type primary-5ess
isdn voice-call-failure 0
call application voice xyz tftp://172.18.16.2/samp/xyz.tcl
call application voice load xys
mta receive maximum-recipients 1024
!
xgcp snmp sgcp
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 framing esf
 clock source line secondary 1
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2
!
controller T1 3
!
voice-port 0:D
!
voice-port 1:D
!
dial-peer voice 4001 pots
 application xyz
 destination-pattern 4003
 port 0:D
 prefix 4001
!
dial-peer voice 513 voip
 destination-pattern 1513200....
 session target ras
!
```

```
dial-peer voice 9002 voip
 destination-pattern 9002
 session target ras
!
dial-peer voice 4191024 pots
 destination-pattern 4192001024
 port 0:D
 prefix 4001
!
dial-peer voice 1513 voip
 destination-pattern 1513.......
 session target ras
!
dial-peer voice 1001 pots
 destination-pattern 14192001001
 port 0:D
!
gateway
 security password 151E0A0E level all
!
 interface Ethernet0
 ip address 10.99.99.7 255.255.255.0
 no ip directed-broadcast
 shutdown
!
interface Serial0:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial1:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 isdn guard-timer 3000
 isdn T203 10000
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 172.18.72.121 255.255.255.192
 no ip directed-broadcast
 duplex auto
 speed auto
 h323-gateway voip interface
 h323-gateway voip id um5300@vgkcisco3 ipaddr 172.18.72.58 1719
 h323-gateway voip h323-id um5300
 h323-gateway voip tech-prefix 1#
!
no ip http server
ip classless
ip route 10.0.0.0 172.18.72.65
!
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
```

```
line aux 0
line vty 0 4
 password xyz
 login
!
ntp clock-period 17179974
ntp server 172.18.72.124
```

H.235 Security Example

The following example shows output from configuring secure registrations from the gatekeeper and identifying which RAS messages the gatekeeper will check to find authentication tokens:

```
dial-peer voice 10 voip
 destination-pattern 4088000
 session target ras
 dtmf-relay h245-alphanumeric
!
gateway
 security password 09404F0B level endpoint
```

The following example shows output from configuring which RAS messages will contain gateway generated tokens:

```
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
radius-server host 10.25.0.0 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server deadtime 5
radius-server key lab
radius-server vsa send accounting
!
gatekeeper
 zone local GK1 test.com 10.0.0.3
 zone remote GK2 test2.com 10.0.2.2 1719
 accounting
 security token required-for registration
 no use-proxy GK1 remote-zone GK2 inbound-to terminal
 no use-proxy GK1 remote-zone GK2 inbound-to gateway
 no shutdown
```

Alternate Gatekeeper Configuration Example

In the following example, the gateway is configured to have alternate gatekeepers. The primary and secondary gatekeepers are configured with the priority option. The priority range is 1 through 127. The first alternate gatekeeper has been configured as priority 120; the second alternate gatekeeper has not been configured, so it remains at the default setting of 127.

```
interface Ethernet 0/1
 ip address 172.18.193.59 255.255.255.0
 h323-gateway voip interface
 h323-gateway voip id GK1 ipaddr 172.18.193.65 1719 priority 120
 h323-gateway voip id GK2 ipaddr 172.18.193.66 1719
 h323-gateway voip h323-id cisco2
```

DTMF Relay Configuration Example

The following example shows DTMF relay configured on a gateway.

```
dial-peer voice 1 voip
 dtmf-relay h245-alphanumeric
 codec g723r53
 destination-pattern 5....
 session target ipv4:192.168.100.2
```

FXS Hookflash Relay Configuration Example

The following example shows how to implement hookflash-in and hookflash-out timing for the hookflash after voice port 1/0/0 has been configured.

```
voice-port 1/0/0
 timing hookflash-in 200
 timing hookflash-out 200
```

Multiple Codec Configuration Example

The following configuration shows how to create a list of prioritized codecs and apply that list to a specific VoIP dial peer:

```
 voice class codec 99
  codec preference 1 g711alaw
  codec preference 2 g711ulaw bytes 80
  codec preference 3 g723ar53
  codec preference 4 g723ar63 bytes 144
  codec preference 5 g723r53
  codec preference 6 g723r63 bytes 120
  codec preference 7 g726r16
  codec preference 8 g726r24
  codec preference 9 g726r32 bytes 80
  codec preference 10 g728
  codec preference 11 g729br8
  codec preference 12 g729r8 bytes 50
!
dial-peer voice 1919 voip
  voice-class codec 99
```

Rotary Calling Pattern Configuration Example

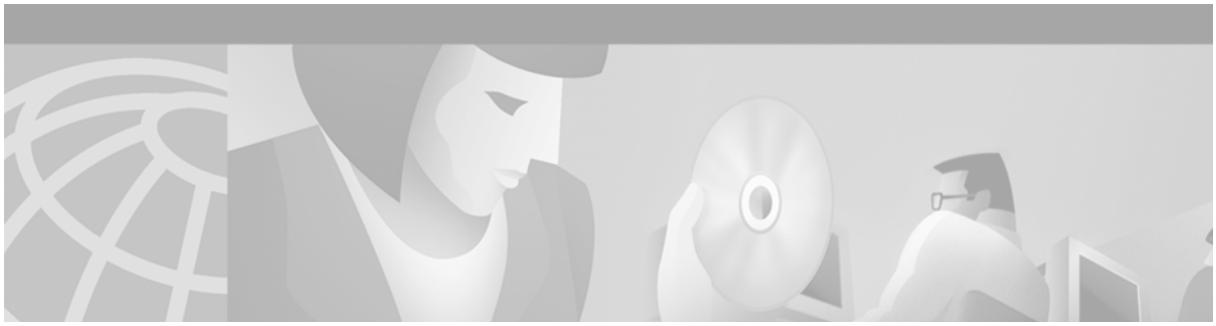
The following example configures POTS dial peer 10 for a preference of 1, POTS dial peer 20 for a preference of 2, and Voice over Frame Relay dial peer 30 for a preference of 3:

```
dial-peer voice 10 pots
 destination pattern 5552150
 preference 1

dial-peer voice 20 pots
 destination pattern 5552150
 preference 2

dial-peer voice 30 vofr
 destination pattern 5552150
 preference 3
```

H.323 Support for Virtual Interfaces Configuration Example

In the following example, Ethernet interface 0/0 is used as the gateway interface. For convenience, the **h323-gateway voip bind srcaddr** command has been specified on the same interface. The designated source IP address is the same as the IP address assigned to the interface.

```
interface Ethernet0/0
 ip address 172.18.194.50 255.255.255.0
 no ip directed-broadcast
 h323-gateway voip interface
 h323-gateway voip id j70f_2600_gk2 ipaddr 172.18.194.53 1719
 h323-gateway voip h323-id j70f_3640_gw1
 h323-gateway voip tech-prefix 3#
 h323-gateway voip bind srcaddr 172.18.194.50
```



Configuring H.323 Gatekeepers and Proxies

This chapter describes how to configure the Cisco Multimedia Conference Manager. The Multimedia Conference Manager provides gatekeeper and proxy capabilities required for service provisioning and management of H.323-compliant networks.

This chapter includes the following sections:

For a complete description of the H.323 gatekeeper commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index or search online.

For more information regarding Resource Reservation Protocol (RSVP), synchronous reservation timers, and slow connect, refer to the Cisco IOS Release 12.1(5)T *VoIP Call Admission Control Using RSVP* or the *Cisco IOS Quality of Service Solutions Configuration Guide.*

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Multimedia Conference Manager Overview

Deploying H.323 applications and services requires careful design and planning for the network infrastructure and for the H.323 devices. The Cisco H.323-compliant Multimedia Conference Manager provides gatekeeper and proxy capabilities, which are required for service provisioning and management of H.323 networks. With the Cisco Multimedia Conference Manager, your current internetwork can be configured to route bit-intensive data, such as audio, telephony, video and audio telephony, and data conferencing using existing telephone and ISDN links without degrading the current level of service of the network. In addition, H.323-compliant applications can be implemented on existing networks in an incremental fashion without upgrades.

Multimedia Conference Manager provides a rich list of networking capabilities, including the following:

- A means to implement quality of service (QoS), which is required for the successful deployment of H.323 applications.
- Interzone routing in the E.164 address space. When using H.323-identification (H.323-ID) format addresses, interzone routing is accomplished by using domain names.

Multimedia Conference Manager allows you to do the following:

- Identify H.323 traffic and apply appropriate policies.
- Limit H.323 traffic on the LAN and WAN.
- Provide user accounting for records based on service use.
- Insert QoS for the H.323 traffic generated by applications such as Voice over IP (VoIP), data conferencing, and video conferencing.
- Implement security for H.323 communications.

Principal Multimedia Conference Manager Functions

The H.323-compliant Multimedia Conference Manager has two principal functions: gatekeeper and proxy. Gatekeeper subsystems provide the following features:

- User authorization in which authentication, authorization, and accounting (AAA) account holders are permitted to register and use the services of the gatekeeper application.
- Accounting using AAA call detail records.
- Zone bandwidth management to limit the number of active sessions.
- H.323 call routing.
- Address resolution.

Cisco Multimedia Conference Managers can be configured to use the Cisco Hot Standby Router Protocol (HSRP) so that when one gatekeeper fails, the standby gatekeeper assumes its role.

Proxy subsystems provide the following features:

- H.323 traffic consolidation.
- Tight bandwidth controls.
- QoS mechanisms such as IP Precedence and RSVP.
- Secure communication over extranets.

H.323 Gatekeeper Features

The following sections describe the main features of a gatekeeper in an H.323 network:

- Zone and Subnet Configuration, page 327
- Redundant H.323 Zone Support, page 327
- Gatekeeper-to-Gatekeeper Redundancy and Load-Sharing Mechanism, page 328
- Interzone Communication, page 329
- RADIUS and TACACS+, page 329
- Accounting via RADIUS and TACACS+, page 329

Zone and Subnet Configuration

A zone is defined as the set of H.323 nodes controlled by a single gatekeeper. Gatekeepers that coexist on a network may be configured so that they register endpoints from different subnets.

Endpoints attempt to discover a gatekeeper and consequently the zone of which they are members by using the Registration, Admission, and Status (RAS) message protocol. The protocol supports a discovery message that may be sent multicast or unicast.

If the message is sent multicast, the endpoint registers nondeterministically with the first gatekeeper that responds to the message. To enforce predictable behavior, where endpoints on certain subnets are assigned to specific gatekeepers, the **zone subnet** command can be used to define the subnets that constitute a given gatekeeper zone. Any endpoint on a subnet that is not enabled for the gatekeeper will not be accepted as a member of that gatekeeper zone. If the gatekeeper receives a discovery message from such an endpoint, it will send an explicit reject message.

Redundant H.323 Zone Support

Redundant H.323 zone support allows for the following:

Gatekeeper Multiple Zone Support

Redundant H.323 zone support allows users to configure multiple remote zones to service the same *zone* or *technology prefix*. A user is able to configure more than one remote gatekeeper to which the local gatekeeper can send location requests (LRQs). This allows for more reliable call completion.

Redundant H.323 zone support is supported on all gatekeeper-enabled IOS images.

Gateway Support for Alternate Gatekeepers

Redundant H.323 zone support in the gateway allows a user to configure two gatekeepers in the gateway (one as the primary and the other as the alternate). All gatekeepers are active. The gateway can choose to register with any one (but not both) at a given time. If that gatekeeper becomes unavailable, the gateway registers with the other.

Redundant H.323 zone support is supported on all gateway-enabled images.

Zone Prefixes

The zone prefixes (typically area codes) serve the same purpose as the domain names in the H.323-ID address space.

For example, the local gatekeeper can be configured with the knowledge that zone prefix "212....." (that is, any address beginning "212" and followed by 7 arbitrary digits) is handled by the gatekeeper gatekeeper_2. Then, when the local gatekeeper is asked to admit a call to destination address 2125551111, it knows to send the LRQ to gatekeeper_2.

When gatekeeper_2 receives the request, the gatekeeper must resolve the address so that the call can be sent to its final destination. There may be an H.323 endpoint with that E.164 address that has registered with gatekeeper_2, in which case gatekeeper_2 returns the IP address for that endpoint. However, it is possible that the E.164 address belongs to a non-H.323 device (for example, a telephone or an H.320 terminal). Because non-H.323 devices do not register with gatekeepers, gatekeeper_2 cannot resolve the address. The gatekeeper must be able to select a gateway that can be used to reach the non-H.323 device. This is where the technology prefixes (or "gateway-type") become useful.

Technology Prefixes

The network administrator selects technology prefixes (tech-prefixes) to denote different types or classes of gateways. The gateways are then configured to register with their gatekeepers with these prefixes. For example, voice gateways can register with tech-prefix 1#, H.320 gateways with tech-prefix 2#, and voicemail gateways with tech-prefix 3#. More than one gateway can register with the same type prefix. When this happens, the gatekeeper makes a random selection among gateways of the same type.

If the callers know the type of device that they are trying to reach, they can include the technology prefix in the destination address to indicate the type of gateway to use to get to the destination. For example, if a caller knows that address 2125551111 belongs to a regular telephone, the destination address of 1#2125551111 can be used, where 1# indicates that the address should be resolved by a voice gateway. When the voice gateway receives the call for 1#2125551111, it strips off the technology prefix and bridges the next leg of the call to the telephone at 2125551111.

Gatekeeper-to-Gatekeeper Redundancy and Load-Sharing Mechanism

The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism expands the capability that is provided by the redundant H.323 zone support feature. Redundant H.323 zone support, which was introduced in Cisco IOS Release 12.1(1)T, allows you to configure multiple gatekeepers to service the same zone or technology prefix by sending LRQs to two or more gatekeepers.

With the redundant H.323 zone support feature, the LRQs are sent simultaneously (in a "blast" fashion) to all of the gatekeepers in the list. The gateway registers with the gatekeeper that responds first. Then, if that gatekeeper becomes unavailable, the gateway registers with another gatekeeper from the list.

The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism allows you to configure gatekeeper support and to give preference to specific gatekeepers. You may choose whether the LRQs are sent simultaneously or sequentially (one at a time) to the remote gatekeepers in the list. If the LRQs are sent sequentially, a *delay* is inserted after the first LRQ and before the next LRQ is sent. This delay allows the first gatekeeper to respond before the LRQ is sent to the next gatekeeper. The order in which LRQs are sent to the gatekeepers is based on the order in which the gatekeepers are listed (using either the **zone prefix** command or the **gw-type-prefix** command).

Once the local gatekeeper has sent LRQs to all the remote gatekeepers in the list (either simultaneously or sequentially), if it has not yet received a location confirmation (LCF), it opens a "window." During this window, the local gatekeeper waits to see whether a LCF is subsequently received from any of the remote gatekeepers. If no LCF is received from any of the remote gatekeepers while the window is open, the call is rejected.

Terminal Name Registration

Gatekeepers recognize one of two types of terminal aliases, or terminal names:

- H.323 IDs, which are arbitrary, case-sensitive text strings
- E.164 addresses, which are telephone numbers

If an H.323 network deploys interzone communication, each terminal should at least have a fully qualified e-mail name as its H.323 identification (ID), for example, bob@cisco.com. The domain name of the e-mail ID should be the same as the configured domain name for the gatekeeper of which it is to be a member. As in the previous example, the domain name would be cisco.com.

Interzone Communication

To allow endpoints to communicate between zones, gatekeepers must be able to determine which zone an endpoint is in and be able to locate the gatekeeper responsible for that zone. If the Domain Name System (DNS) mechanism is available, a DNS domain name can be associated with each gatekeeper. See the DNS configuration task in the "Configuring Intergatekeeper Communication" section to understand how to configure DNS.

RADIUS and TACACS+

Version 1 of the H.323 specification does not provide a mechanism for authenticating registered endpoints. Credential information is not passed between gateways and gatekeepers. However, by enabling AAA on the gatekeeper and configuring for RADIUS and TACACS+, a rudimentary form of identification can be achieved.

If the AAA feature is enabled, the gatekeeper attempts to use the registered aliases along with a password and completes an authentication transaction to a RADIUS and TACACS+ server. The registration will be accepted only if RADIUS and TACACS+ successfully authenticates the name.

The gatekeeper can be configured so that a default password can be used for all users. The gatekeeper can also be configured so that it recognizes a password separator character that allows users to piggyback their passwords onto H.323-ID registrations. In this case, the separator character separates the ID and password fields.

Note The names loaded into RADIUS and TACACS+ are probably not the same names provided for dial access because they may all have the same password.

Accounting via RADIUS and TACACS+

If AAA is enabled on the gatekeeper, the gatekeeper will emit an accounting record each time a call is admitted or disconnected.

Interzone Routing Using E.164 Addresses

Interzone routing may be configured using E.164 addresses.

Two types of address destinations are used in H.323 calls. The destination can be specified using either an H.323-ID address (a character string) or an E.164 address (a string that contains telephone keypad characters). The way interzone calls are routed depends on the type of address being used.

When using H.323-ID addresses, interzone routing is handled through the use of domain names. For example, to resolve the domain name bob@cisco.com, the source endpoint gatekeeper finds the gatekeeper for cisco.com and sends it the location request for the target address bob@cisco.com. The destination gatekeeper looks in its registration database, sees bob registered, and returns the appropriate IP address to get to bob.

When using E.164 addresses, call routing is handled through zone prefixes and gateway-type prefixes, also referred to as technology prefixes. The zone prefixes, which are typically area codes, serve the same purpose as domain names in H.323-ID address routing. Unlike domain names, however, more than one zone prefix can be assigned to one gatekeeper, but the same prefix cannot be shared by more than one gatekeeper.

Use the **zone prefix** command to define gatekeeper responsibilities for area codes. The command can also be used to tell the gatekeeper which prefixes are in its own zones and which remote gatekeepers are responsible for other prefixes.

Note Area codes are used as an example in this section, but a zone prefix need not be an area code. It can be a country code, an area code plus local exchange (NPA-NXX), or any other logical hierarchical partition.

The following sample command shows how to configure a gatekeeper with the knowledge that zone prefix 212..... (that is, any address beginning with area code 212 and followed by seven arbitrary digits) is handled by gatekeeper gk-ny:

```
my-gatekeeper(config-gk)# zone prefix gk-ny 212.......
```

When my-gatekeeper is asked to admit a call to destination address 2125551111, it knows to send the location request to gk-ny.

However, once the query gets to gk-ny, gk-ny still needs to resolve the address so that the call can be sent to its final destination. There could be an H.323 endpoint that has registered with gk-ny with that E.164 address, in which case gk-ny would return the IP address for that endpoint. However, it is more likely that the E.164 address belongs to a non-H.323 device, such as a telephone or an H.320 terminal.

Because non-H.323 devices do not register with gatekeepers, gk-ny has no knowledge of which device the address belongs to or which type of device it is, so the gatekeeper cannot decide which gateway should be used for the *hop off* to the non-H.323 device. (The term *hop off* refers to the point at which the call leaves the H.323 network and is destined for a non-H.323 device.)

Note The number of zone prefixes defined for a directory gatekeeper that is dedicated to forwarding LRQs, and not for handling local registrations and calls, should not exceed 10,000; 4 MB of memory must be dedicated to describing zones and zone prefixes to support this maximum number of zone prefixes. The number of zone prefixes defined for a gatekeeper that handles local registrations and calls should not exceed 2000.

To enable the gatekeeper to select the appropriate hop-off gateway, use the **gw-type-prefix** command to configure technology or gateway-type prefixes. Select technology prefixes to denote different types or classes of gateways. The gateways are then configured to register with their gatekeepers using these technology prefixes.

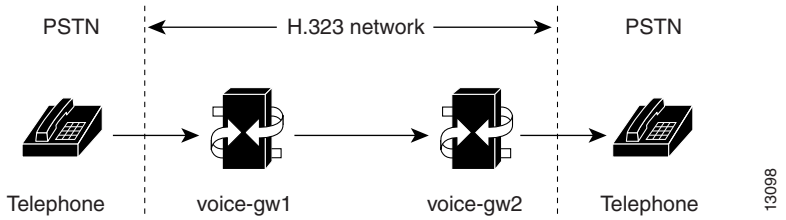
For example, voice gateways might register with technology prefix 1#, and H.320 gateways might register with technology prefix 2#. If there are several gateways of the same type, configure them to register with the same prefix type. By having them register with the same prefix type, the gatekeeper treats the gateways as a pool out of which a random selection is made whenever a call for that prefix type arrives. If a gateway can serve more than one type of hop-off technology, it can register more than one prefix type with the gatekeeper.

Callers will need to know the technology prefixes that are defined. The callers will need to know the type of device they are trying to reach and will need to prepend the appropriate technology prefix to the destination address to indicate the type of gateway needed to reach the destination.

For example, callers might request 1#2125551111 if they know that address 2125551111 is for a telephone and that the technology prefix for voice gateways is 1#. The voice gateway is configured with a dial peer (using the **dial-peer** command) so that when the gateway receives the call for 1#2125551111, it strips off the technology prefix 1# and bridges the next leg of the call to the telephone at 2125551111.

In cases in which the call scenario is as shown in Figure 74, voice-gw1 can be configured to prepend the voice technology prefix 1# so that the use of technology prefixes is completely transparent to the caller.

Figure 74 Call Scenario



Additionally, in using the **gw-type-prefix** command, a particular gateway-type prefix can be defined as the default gateway type to be used for addresses that cannot be resolved. It also forces a technology prefix to always hop off in a particular zone.

If the majority of calls hop off on a particular type of gateway, the gatekeeper can be configured to use that type of gateway as the default type so that callers no longer have to prepend a technology prefix on the address. For example, if voice gateways are mostly used in a network, and all voice gateways have been configured to register with technology prefix 1#, the gatekeeper can be configured to use 1# gateways as the default technology if the following command is entered:

```
my-gatekeeper(config-gk)# gw-type-prefix 1# default-technology
```

Now a caller no longer needs to prepend 1# to use a voice gateway. Any address that does not contain an explicit technology prefix will be routed to one of the voice gateways that registered with 1#.

With this default technology definition, a caller could ask the gatekeeper for admission to 2125551111. If the local gatekeeper does not recognize the zone prefix as belonging to any remote zone, it will route the call to one of its local (1#) voice gateways so that the call hops off locally. However, if it knows that gk-ny handles the 212 area code, it can send a location request for 2125551111 to gk-ny. This requires that gk-ny also be configured with some default gateway type prefix and that its voice gateways be registered with that prefix type.

Note For ease of maintenance, the same prefix type should be used to denote the same gateway type in all zones under your administration. No more than 50 different technology prefixes should be registered per zone.

Also, with the **gw-type-prefix** command, a hop off can be forced to a particular zone. When an endpoint or gateway makes a call-admission request to its gatekeeper, the gatekeeper determines the destination address by first looking for the technology prefix. When that is matched, the remaining string is compared against known zone prefixes. If the address is determined to be a remote zone, the entire address, including technology and zone prefixes, is sent to the remote gatekeeper in a location request. That remote gatekeeper then uses the technology prefix to decide on which of its gateways to hop off. In other words, the zone prefix (defined using the **zone prefix** command) determines the routing to a zone, and once there, the technology prefix (defined using the **gw-type-prefix** command) determines the gateway to be used in that zone. The zone prefix takes precedence over the technology prefix.

This behavior can be overridden by associating a forced hop-off zone with a particular technology prefix. Associating a forced hop-off zone with a particular technology prefix forces the call to the specified zone, regardless of what the zone prefix in the address is. As an example, you are in the 408 area code and want callers to the 212 area code in New York to use H.323-over-IP and hop off there because it saves on costs. However, the only H.320 gateway is in Denver. In this example, calls to H.320 endpoints must be forced to hop off in Denver, even if the destination H.320 endpoint is in the 212 area code. The forced hop-off zone can be either a local zone (that is, one that is managed by the local gatekeeper) or a remote zone.

HSRP Support

Cisco routers support Hot Standby Router Protocol (HSRP), which allows one router to serve as a backup to another router. Cisco gatekeepers can be configured to use HSRP so that when one gatekeeper fails, the standby gatekeeper assumes its role.

To configure a gatekeeper to use HSRP, perform the following tasks:

- Select one interface on each gatekeeper to serve as the HSRP interface and configure these two interfaces so that they belong to the same HSRP group but have different priorities. The one with the higher priority will be the active gatekeeper; the other assumes the standby role. Make a note of the virtual HSRP IP address shared by both of these interfaces. (For details on HSRP and HSRP configuration, refer to the *Cisco IOS IP Configuration Guide*.)
- Configure the gatekeepers so that the HSRP virtual IP address is the RAS address for all local zones.
- Make sure that the gatekeeper-mode configurations on both routers are identical.
- If the endpoints and gateways are configured so that they use a specific gatekeeper address (rather than multicasting), use the HSRP virtual IP address as the gatekeeper address. You can also let the endpoints and gateways find the gatekeeper by multicasting. As long as it is on standby status, the secondary gatekeeper neither receives nor responds to multicast or unicast requests.

As long as both gatekeepers are up, the one with the higher priority on its HSRP interface will be the active gatekeeper. If this active gatekeeper fails, or if its HSRP interface fails, the standby HSRP interface assumes the virtual HSRP address and, with it, the active gatekeeper role. When the gatekeeper with the higher HSRP priority comes back online, it reclaims the HSRP virtual address and the gatekeeper function, while the secondary gatekeeper goes back to standby status.

Note Gatekeeper failover will not be completely transparent to endpoints and gatekeepers. When the standby gatekeeper takes over, it does not have the state of the failed gatekeeper. If an endpoint that had registered with the failed gatekeeper now makes a request to the new gatekeeper, the gatekeeper responds with a reject, indicating that it does not recognize the endpoint. The endpoint must reregister with the new gatekeeper before it can continue H.323 operations.

For an example of configuring gatekeeper HSRP support, see the "H.323 Gatekeeper and Proxy Configuration Examples" section.

H.323 Proxy Features

Each of the following sections describes how the proxy feature can be used in an H.323 network:

- Security, page 333
- Quality of Service, page 337
- Application-Specific Routing, page 337

Security

When terminals signal each other directly, they must have direct access to each other's addresses. This exposes an attacker to key information about a network. When a proxy is used, the only addressing information that is exposed to the network is the address of the proxy; all other terminal and gateway addresses are hidden.

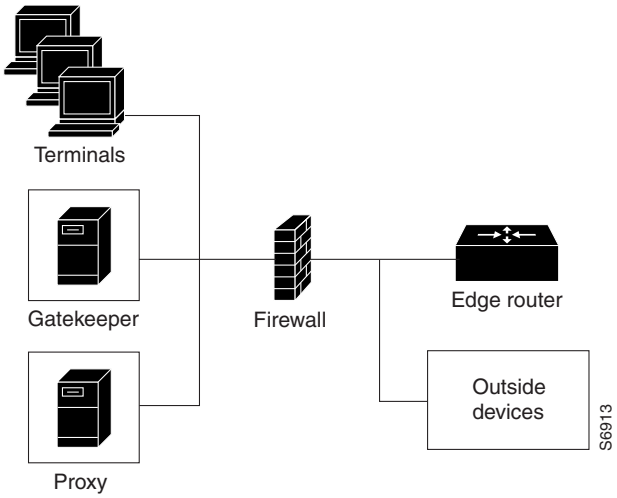
There are several ways to use a proxy with a firewall to enhance network security. The configuration to be used depends on how capable the firewall is of handling the complex H.323 protocol suite. Each of the following sections describes a common configuration for using a proxy with a firewall:

- Proxy Inside the Firewall, page 334
- Proxy in Co-Edge Mode, page 335
- Proxy Outside the Firewall, page 336
- Proxies and NAT, page 336

Proxy Inside the Firewall

H.323 is a complex, dynamic protocol that consists of several interrelated subprotocols. During H.323 call setup, the ports and addresses released with this protocol require a detailed inspection as the setup progresses. If the firewall does not support this dynamic access control based on the inspection, a proxy can be used just inside the firewall. The proxy provides a simple access control scheme, as illustrated in Figure 75.

Figure 75 Proxy Inside the Firewall

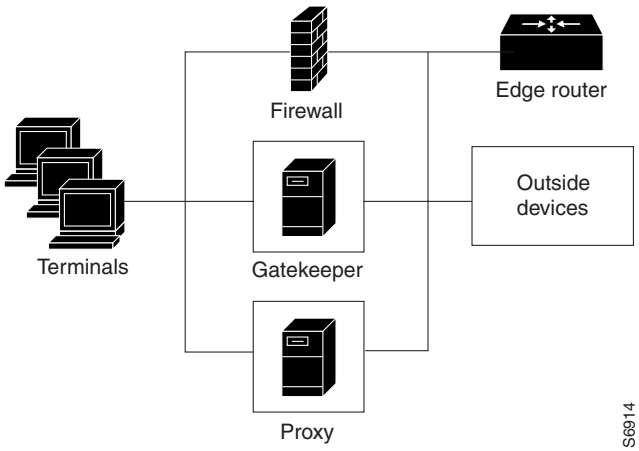


Because the gatekeeper (using RAS) and the proxy (using call setup protocols) are the only endpoints that communicate with other devices outside the firewall, it is simple to set up a tunnel through the firewall to allow traffic destined for either of these two endpoints to pass through.

Proxy in Co-Edge Mode

If H.323 terminals exist in an area with local interior addresses that must be translated to valid exterior addresses, the firewall must be capable of decoding and translating all addresses passed in the various H.323 protocols. If the firewall is not capable of this translation task, a proxy may be placed next to the firewall in a co-edge mode. In this configuration, interfaces lead to both inside and outside networks. (See Figure 76.)

Figure 76 Proxy in Co-Edge Mode



In co-edge mode, the proxy can present a security risk. To avoid exposing a network to unsolicited traffic, configure the proxy to route only proxied traffic. In other words, the proxy routes only H.323 protocol traffic that is terminated on the inside and then repeated to the outside. Traffic that moves in the opposite direction can be configured this way as well.

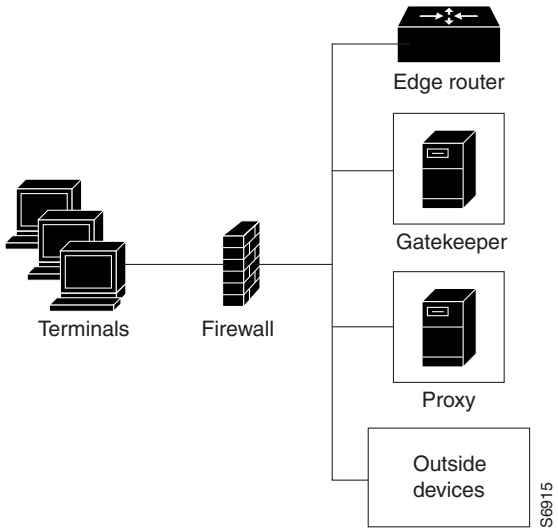
Proxy Outside the Firewall

To place the proxy and gatekeeper outside the firewall, two conditions must exist. First, the firewall must support H.323 dynamic access control. Second, Network Address Translation (NAT) must not be in use.

If NAT is in use, each endpoint must register with the gatekeeper for the duration of the time it is online. This will quickly overwhelm the firewall because a large number of relatively static, internal-to-external address mappings will need to be maintained.

If the firewall does not support H.323 dynamic access control, the firewall can be configured with static access lists that allow traffic from the proxy or gatekeeper through the firewall. This can present a security risk if an attacker can *spoof*, or simulate, the IP addresses of the gatekeeper or proxy and use them to attack the network. Figure 77 illustrates proxy outside the firewall.

Figure 77 Proxy Outside the Firewall



Proxies and NAT

When a firewall is providing NAT between an internal and an external network, proxies may allow H.323 traffic to be handled properly, even in the absence of a firewall that can translate addresses for H.323 traffic. Table 27 and Table 28 provide guidelines for proxy deployment for networks that use NAT.

Table 27 Guidelines for Networks That Use NAT

For Networks Using NAT	Firewall with H.323 NAT	Firewall Without H.323 NAT
Firewall with dynamic access control	Gatekeeper and proxy inside the firewall	Co-edge gatekeeper and proxy
Firewall without dynamic access control	Gatekeeper and proxy inside the firewall, with static access lists on the firewall	Co-edge gatekeeper and proxy

Table 28 Guidelines for Networks That Do Not Use NAT

For Networks Not Using NAT	Firewall with H.323. NAT	Firewall Without H.323 NAT
Firewall with Dynamic Access Control	Gatekeeper and proxy inside the firewall Gatekeeper and proxy outside the firewall	Gatekeeper and proxy inside the firewall Gatekeeper and proxy outside the firewall
Firewall Without Dynamic Access Control	Gatekeeper and proxy inside the firewall, with static access lists on the firewall	Gatekeeper and proxy inside the firewall, with static access lists on the firewall

Quality of Service

Quality of service (QoS) enables complex networks to control and predictably service a variety of applications. QoS expedites the handling of mission-critical applications while sharing network resources with noncritical applications. QoS also ensures available bandwidth and minimum delays required by time-sensitive multimedia and voice applications. In addition, QoS gives network managers control over network applications, improves cost-efficiency of WAN connections, and enables advanced differentiated services. QoS technologies are elemental building blocks for other Cisco IOS-enabling services such as its H.323-compliant gatekeeper. Overall call quality can be improved dramatically in the multimedia network by using pairs of proxies between regions of the network where QoS can be requested.

When two H.323 terminals communicate directly, the resulting call quality can range from good (for high-bandwidth intranets) to poor (for most calls over the public network). As a result, deployment of H.323 is almost always predicated on the availability of some high-bandwidth, low-delay, low-packet-loss network that is separate from the public network or that runs overlaid with the network as a premium service and adequate QoS.

Adequate QoS usually requires terminals that are capable of signaling such premium services. There are two major ways to achieve such signaling:

- RSVP to reserve flows having adequate QoS based on the media codecs of H.323 traffic
- IP precedence bits to signal that the H.323 traffic is special and that it deserves higher priority

Unfortunately, the vast majority of H.323 terminals cannot achieve signaling in either of these ways.

The proxy can be configured to use any combination of RSVP and IP precedence bits.

The proxy is not capable of modifying the QoS between the terminal and itself. To achieve the best overall QoS, ensure that terminals are connected to the proxy using a network that intrinsically has good QoS. In other words, configure a path between a terminal and proxy that provides good bandwidth, delay, and packet-loss characteristics without the terminal needing to request special QoS. A high-bandwidth LAN works well for this.

Application-Specific Routing

To achieve adequate QoS, a separate network may be deployed that is partitioned away from the standard data network. The proxy can take advantage of such a partitioned network using a feature known as application-specific routing (ASR).

Application-specific routing is simple. When the proxy receives outbound traffic, it directs traffic to an interface that is connected directly to the QoS network. The proxy does not send the traffic using an interface that is specified for the regular routing protocol. Similarly, inbound traffic from other proxies is received on the interface that is connected to the QoS network. This is true if all these other proxies around the QoS network use ASR in a consistent fashion. ASR then ensures that ordinary traffic is not routed into the QoS network by mistake.

Implementation of ASR ensures the following:

- Each time a connection is established with another proxy, the proxy automatically installs a host route pointing at the interface designated for ASR.
- The proxy is configured to use a loopback interface address. The proxy address is visible to both the ASR interface and all regular interfaces, but there are no routes established between the loopback interface and the ASR interface. This ensures that no non-H.323 traffic is routed through the ASR interface.

Note ASR is not supported on Frame Relay or ATM interfaces for the Cisco MC3810 platform.

H.323 Prerequisite Tasks and Restrictions

This section contains prerequisite tasks and restrictions for configuring H.323 gatekeepers and proxies.

Redundant H.323 Zone Support

Redundant H.323 zone support has the following restrictions and limitations:

- The gateway can register with only one gatekeeper at any given time.
- Only E.164 address resolution is supported.
- Because the gateway can register with only one gatekeeper at a time, redundant H.323 zone support provides only redundancy and does not provide any load balancing.
- Although redundant H.323 zone support allows you to configure alternate gatekeepers, it will not insert information in the alternate gatekeeper field of some RAS messages.

Gatekeeper-to-Gatekeeper Redundancy and Load-Sharing Mechanism

The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism has the following restrictions and limitations:

- The gatekeeper-to-gatekeeper redundancy and load-sharing mechanism requires the Cisco H.323 VoIP Gatekeeper for Cisco Access Platforms feature.
- The order in which LRQs are sent to the gatekeepers is based on the order in which the gatekeepers are listed. You cannot specify a priority number for a gatekeeper.
- Regardless of the order in which the LRQs are sent, the gateway will still use the first gatekeeper that sends an LCF.
- The settings for delay between LRQs and the LRQ window are global and cannot be set on a per-zone or technology-prefix basis.

- The number of remote gatekeepers multiplied by the delay per LRQ cannot exceed the Routing Information Protocol (RIP) timeout. Therefore, we recommend that you limit your list of remote gatekeepers to two or three.
- If LRQ forwarding is enabled on the directory gatekeeper, the *sequential* setting for LRQs is ignored.
- Only E.164 address resolution is supported.
- Using redundant H.323 zone support in the "directory gatekeeper" can generate extra RAS messages. Therefore, the number of "directory gatekeeper" levels should be kept to a minimum (two or three at the maximum).

H.323 Gatekeeper Configuration Task List

To configure Cisco gatekeepers, perform the tasks in the following sections. The tasks in these two sections are required.

- Configuring the Gatekeeper, page 339 (Required)
- Configuring the Proxy, page 368 (Required)

Configuring the Gatekeeper

To configure gatekeepers, perform the tasks in the following sections. All of the tasks listed are required.

- Starting a Gatekeeper, page 340
  - Configuring Intergatekeeper Communication, page 343
- Configuring Redundant H.323 Zone Support, page 344
- Configuring Local and Remote Gatekeepers, page 345
- Configuring Redundant Gatekeepers for a Zone Prefix, page 346
- Configuring Redundant Gatekeepers for a Technology Prefix, page 347
- Configuring Static Nodes, page 349
- Configuring H.323 Users via RADIUS, page 350
- Configuring a RADIUS/AAA Server, page 354
- Configuring User Accounting Activity for RADIUS, page 356
- Configuring E.164 Interzone Routing, page 357
- Configuring H.323 Version 2 Features, page 358
  - Configuring a Dialing Prefix for Each Gateway, page 359
  - Configuring a Prefix to a Gatekeeper Zone List, page 362
  - Configuring a Gatekeeper for Interaction with External Applications, page 361
  - Configuring Gatekeeper Triggers for Interaction with External Applications, page 363
  - Configuring Redundant H.323 Zone Support, page 344
  - Configuring a Forced Disconnect on a Gatekeeper, page 368

Starting a Gatekeeper

To enter gatekeeper configuration mode and to start the gatekeeper, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. The arguments are as follows: • *gatekeeper-name*—Specifies the gatekeeper name or zone name. This is usually the fully domain-qualified host name of the gatekeeper. For example, if the domain name is cisco.com, the gatekeeper name might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the gatekeeper name for each zone should be some unique string that has a mnemonic value. • *domain-name*—Specifies the domain name served by this gatekeeper. • *ras-IP-address*—(Optional) Specifies the IP address of one of the interfaces on the gatekeeper. When the gatekeeper responds to gatekeeper discovery messages, it signals the endpoint or gateway to use this address in future communications. **Note** Setting this address for one local zone makes it the address used for all local zones.

Command	Purpose
Step 3 `Router(config-gk)# zone prefix gatekeeper-name e164-prefix [blast | seq] [gw-priority priority gw-alias [gw-alias, ...]]`	Adds a prefix to the gatekeeper zone list. The keywords and arguments are as follows: • *gatekeeper-name*—Specifies the name of a local or remote gatekeeper, which must have been defined by using the **zone local** or **zone remote** command. • *e164-prefix*—Specifies an E.164 prefix in standard form followed by dots (.). Each dot represents a number in the E.164 address. For example, 212..... is matched by 212 and any 7 numbers. **Note** Although a dot to represent each digit in an E.164 address is the preferred configuration method, you can also enter an asterisk (*) to match any number of digits. • **blast**—(Optional) If you list multiple hopoffs, indicates that the location requests (LRQs) should be sent simultaneously to the gatekeepers based on the order in which they were listed. The default is seq. • **seq**—(Optional) If you list multiple hopoffs, indicates that the LRQs should be sent sequentially to the gatekeepers based on the order in which they were listed. The default is seq. • **gw-priority** *priority gw-alias*—(Optional) Use the **gw-priority** option to define how the gatekeeper selects gateways in its local zone for calls to numbers that begin with prefix e164-prefix. Do not use this option to set priority levels for a prefix assigned to a remote gatekeeper. Use values from 0 to 10. A 0 value prevents the gatekeeper from using the gateway *gw-alias* for that prefix. Value 10 places the highest priority on gateway *gw-alias*. If you do not specify a priority value for a gateway, the value 5 is assigned. To assign the same priority value for one prefix to multiple gateways, list all the gateway names after the pri-0-to-10 value. The *gw-alias* name is the H.323 ID of a gateway that is registered or will register with the gatekeeper. This name is set on the gateway with the **h323-gateway voip h.323-id** command.

	Command	Purpose
Step 4	`Router(config-gk)# zone subnet local-gatekeeper-name [default | subnet-address {/bits-in-mask | mask-address} enable]`	Defines a set of subnets that constitute the gatekeeper zone. Enables the gatekeeper for each of these subnets and disables it for all other subnets. (Repeat for all subnets.) The keywords and arguments are as follows: • *local-gatekeeper-name*—Specifies the name of the local gatekeeper. • **default**—(Optional) Applies to all other subnets that are not specifically defined by the **zone subnet** command. • *subnet-address*—(Optional) Specifies the address of the subnet that is being defined. • *bits-in-mask*—(Optional) Specifies the number of bits of the mask to be applied to the subnet address. **Note** The slash must be entered before this argument. • *mask-address*—(Optional) Specifies the mask (in dotted string format) to be applied to the subnet address. • **enable**—(Optional) Specifies that the gatekeeper accepts discovery and registration from the specified subnets. **Note** To define the zone as being all but one set of subnets by disabling that set and enabling all other subnets, use the **no** form of the command as follows: Configure **no zone subnet** *local-gatekeeper-name subnet-address* {*/bits-in-mask* | *mask-address*} **enable**. **Note** To accept the default behavior, which is that all subnets are enabled, use the **no** form of the command as follows: **no zone subnet** *local-gatekeeper-name* **default enable**.
Step 5	`Router(config-gk)# no shutdown`	Brings the gatekeeper online.

The *local-gatekeeper-name* argument should be a Domain Name System (DNS) host name if DNS is to be used to locate remote zones.

The **zone subnet** command may be used more than once to create a list of subnets controlled by a gatekeeper. The subnet masks need not match actual subnets in use at your site. For example, to specify a particular endpoint, show its address as a 32-bit netmask.

If a local gatekeeper name is contained in the message, it must match the *local-gatekeeper-name* argument.

Note To explicitly enable or disable a particular endpoint, specify its host address using a 32-bit subnet mask.

Configuring Intergatekeeper Communication

This section describes two ways to configure intergatekeeper communication:

- Via DNS, page 343
- Manual Configuration, page 344

Via DNS

To configure intergatekeeper communication using DNS, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip name-server dns-server-name [server-address2...server-address6]`	Specifies the DNS server address. The arguments are as follows: • *dns-server-name*— Specifies the IP address of the name server. • *server-address2...server-address6*—(Optional) IP addresses of additional name servers (a maximum of six name servers).
Step 2	`Router(config)# ip domain-name name`	Defines a default domain name that the Cisco IOS software uses to complete unqualified host names (names without a dotted-decimal domain name). The *name* argument specifies the default domain name used to complete unqualified host names. Do not include the initial period that separates an unqualified name from the domain name.

For all gatekeepers in the system, enter a text record of the form into DNS:

ras [*gk-id@*] *host* [*:port*] [*priority*]

The *gk-id* argument is an optional gatekeeper ID. If the optional gatekeeper ID is not specified, *host* is used as the gatekeeper ID.

The *host* argument is either an IP address or the actual host name of the gatekeeper in the form *host.some_domain.com*.

The *port* argument, if specified, should be some port number other than RAS port 1719.

The *priority* argument specifies the order in which the listed gatekeepers should be searched for endpoints. Gatekeepers with lower priorities are searched before those with higher numbers.

How you enter the text record for a particular domain depends on the DNS implementation. The following examples are for the Berkeley Internet Name Domain (BIND). These records are typically entered into the “hosts” database:

zone1.comintxt“ras gk.zone1.com”
zone2.comintxt“ras gk2@gk.zone2.com”

zone3.comintxt“ras gk.3@gk.zone3.com:1725”
zone4.comintxt“ras gk4@gk.zone4.com:1725 123”
zone5.comintxt“ras gk5@101.0.0.1:1725”

Manual Configuration

If you choose not to use DNS or if DNS is not available, configure intergatekeeper communication manually. To configure intergatekeeper manual communication, use the following command in gatekeeper configuration mode for every other gatekeeper in the network:

Command	Purpose
`Router(config-gk)# zone remote` *other-gatekeeper-name other-domain-name other-gatekeeper-ip-address* [*port-number*]	Statically specifies a remote zone if Domain Name System (DNS) is unavailable or undesirable. Enter this command for each gatekeeper. The arguments are as follows: • *other-gatekeeper-name*—Specifies the name of the remote gatekeeper. • *other-domain-name*—Specifies the domain name of the remote gatekeeper. • *other-gatekeeper-ip-address*—Specifies the IP address of the remote gatekeeper. • *port-number*—(Optional) Specifies the RAS signaling port number for the remote zone. Value ranges are from 1 to 65,535. If this option is not set, the default is the well-known RAS port number 1719.

Configuring Redundant H.323 Zone Support

Regardless of whether you specify sequential or blast, there is an order to how the LRQs are sent. With sequential, the LRQs are sent one at a time with a delay between each. With blast, the LRQs are sent back-to-back in a rapid sequence without any delay between them. The order in which zone and technology prefixes are configured determines the order in which the LRQs are sent to the remote gatekeepers. Using zone prefixes as an example, the local gatekeeper routes the call to the first zone that responds with an LCF. If the local gatekeeper is configured for a zone prefix that already has remote gatekeepers configured, the local gatekeeper will automatically put that zone prefix at the top of the list.

For example:

```
gatekeeper
 zone local gnet-2503-2-gk cisco.com
 zone remote gnet-2600-1-gk cisco.com 172.18.194.131 1719
 zone remote gnet-2503-3-gk cisco.com 172.18.194.134 1719
 zone prefix gnet-2600-1-gk 919.......
 zone prefix gnet-2503-6-gk 919.......
```

With this configuration, LRQs are first sent to gnet-2600-1-gk (which is the first zone prefix because it has a remote gatekeeper configured for it) and then to gnet-2503-6-gk (which is the second zone prefix). If you add the local gatekeeper to that zone prefix, it automatically goes to the top of the list, as shown below:

```
gatekeeper
 zone local gnet-2503-2-gk cisco.com
 zone remote gnet-2600-1-gk cisco.com 172.18.194.131 1719
```

```
zone remote gnet-2503-3-gk cisco.com 172.18.194.134 1719
zone prefix gnet-2503-2-gk 919.......
zone prefix gnet-2600-1-gk 919.......
zone prefix gnet-2503-6-gk 919.......
```

As you can see, the zone prefix for the local gatekeeper (gnet-2503-2-gk) has been inserted at the top of the zone prefix list. If the local gatekeeper can resolve the address, it will not send LRQs to the remote zones.

If you are configuring technology prefixes, the zone prefix for the local gatekeeper should be inserted at the top of the zone prefix list. If the local gatekeeper can resolve the address, it will not send LRQs to the remote zones.

Configuring Local and Remote Gatekeepers

To configure local and remote gatekeepers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. The arguments are as follows: • *gatekeeper-name*—Specifies the gatekeeper name or zone name. This is usually the fully domain-qualified host name of the gatekeeper. For example, if the domain name is cisco.com, the gatekeeper name might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the gatekeeper name for each zone should be some unique string that has a mnemonic value. • *domain-name*—Specifies the domain name served by this gatekeeper. • *ras-IP-address*—(Optional) Specifies the IP address of one of the interfaces on the gatekeeper. When the gatekeeper responds to gatekeeper discovery messages, it signals the endpoint or gateway to use this address in future communications. **Note** Setting this address for one local zone makes it the address used for all local zones.

	Command	Purpose
Step 3	`Router(config-gk)# zone remote other-gatekeeper-name other-domain-name other-gatekeeper-ip-address [port-number]`	Configures the remote gatekeeper. The arguments are as follows: • *other-gatekeeper-name*—Name of the remote gatekeeper. • *other-domain-name*—Domain name of the remote gatekeeper. • *other-gatekeeper-ip-address*—IP address of the remote gatekeeper. • *port-number*—(Optional) RAS signaling port number for the remote zone. Value ranges from 1 to 65,535. If this option is not set, the default is the well-known RAS port number 1719.

Configuring Redundant Gatekeepers for a Zone Prefix

To configure redundant gatekeepers for a zone prefix, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone prefix gatekeeper-name e164-prefix [blast | seq] [gw-priority priority gw-alias [gw-alias, ...]]`	Adds a prefix to the gatekeeper zone list. For an explanation of the keywords and arguments, see Step 3 of the configuration task table in the "Starting a Gatekeeper" section on page 340.

You can configure multiple remote gatekeepers for the same prefix, but only one of the gatekeepers defined for any given zone prefix can be local. It is recommended that you limit the number of remote gatekeepers that service the same zone prefix to two.

By default, LRQs are sent sequentially to the remote gatekeepers. If you would like the LRQs to be sent simultaneously (blast), you need only specify the **blast** keyword on one **zone prefix** command per E.164 prefix.

Verifying Zone Prefix Redundancy

To verify the order in which LRQs will be sent to the gatekeepers defined for a zone prefix, enter the **show gatekeeper zone prefix** command. The following output lists all the gatekeepers, in order, and the zone prefixes serviced by each.

```
router# show gatekeeper zone prefix

ZONE PREFIX TABLE
      =================
GK-NAME               E164-PREFIX
-------               -----------
c3620-1-gk            917300....
c2514-2-gk            917300....
c2600-1-gk            919.......
c2514-1-gk            919.......
```

To verify whether the LRQs will be sent sequentially or simultaneously to the gatekeepers, enter the **show running-config** command. If the LRQs will be sent simultaneously, blast will appear beside the first entry for a particular zone (as shown in the following output for zone 919).

```
Router# show running-config

Building configuration...

Current configuration:
!
gatekeeper
 zone remote c3620-1-gk cisco.com 172.18.194.79 1719
 zone remote c2514-2-gk cisco.com 172.18.194.89 1719
 zone remote gk-cisco-paul cisco.com 172.18.193.155 1719
 zone prefix c3620-1-gk 917300....
 zone prefix c2514-2-gk 917300....
 zone prefix c2514-2-gk 919..... blast
 zone prefix c3620-1-gk 919.....
```

Configuring Redundant Gatekeepers for a Technology Prefix

To configure redundant gatekeepers for a technology prefix, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# gw-type-prefix` *type-prefix* `[[hopoff` *gkid1*`] [hopoff` *gkid2*`] [hopoff` *gkidn*`] [seq | blast]] [default-technology] [[gw ipaddr` *ipaddr* `[`*port*`]]...]`	Configures the gatekeepers to service a technology zone and specifies whether LRQs should be sent in blast or sequential fashion. The default is sequential. The keywords and arguments are as follows: • *type-prefix*—Specifies that a technology prefix is recognized and stripped before checking for the zone prefix. It is strongly recommended that you select technology prefixes that do not lead to ambiguity with zone prefixes. Do this by using the # character to terminate technology prefixes, for example, 3#. • **hopoff** *gkid*—(Optional) Specifies the gatekeeper where the call is to hop off, regardless of the zone prefix in the destination address. The *gkid* argument refers to a gatekeeper previously configured using the **zone local** or **zone remote** command. You can enter this keyword and argument multiple times to configure redundant gatekeepers for a given technology prefix.

Command	Purpose
	• **seq | blast**—(Optional) If multiple hopoffs are listed, indicates that the location requests (LRQs) should be sent sequentially or simultaneously (blast) to the gatekeepers based on the order in which they were listed. The default is to send them sequentially. • **default-technology**—(Optional) Specifies that gateways that register with this prefix option are used as the default for routing any addresses that are otherwise unresolved. • **gw ipaddr** *ipaddr* [*port*]—(Optional) Indicates that the gateway is incapable of registering technology prefixes. When it registers, it adds the gateway to the group for this type-prefix, just as if it had sent the technology prefix in its registration. This parameter can be repeated to associate more than one gateway with a technology prefix.

You can enter the **hopoff** keyword and *gkid* argument multiple times in the same command to define a group of gatekeepers that will service a given technology prefix. After you have listed all of the gatekeepers that will service that technology zone, you can specify whether the LRQs should be sent in blast or sequential fashion.

Note Only one of the gatekeepers in the hopoff list can be local. We recommend that you limit the number of remote gatekeepers that service the same technology prefix to two.

Verifying Technology Prefix Redundancy

To verify that multiple gatekeepers are defined for a technology prefix, enter the **show gatekeeper gw-type-prefix** command. The following output displays the gateway technology prefix table.

```
router# show gatekeeper gw-type-prefix

(GATEWAYS-TYPE PREFIX TABLE
==============================
Prefix:3#*    (Hopoff zone c2600-1-gk c2514-1-gk)
```

To verify whether the LRQs will be sent sequentially or simultaneously to the gatekeepers, enter the **show running-config** command. If the LRQs will be sent simultaneously, blast will appear at the end of the gw-type-prefix line (as shown below).

```
Router# show running-config

Building configuration...

Current configuration:
!
gatekeeper
 zone remote c2600-1-gk cisco.com 172.18.194.70 1719
 zone remote c2514-1-gk cisco.com 172.18.194.71 1719
 gw-type-prefix 3#* hopoff c2600-1-gk hopoff c2514-1-gk blast
```

Configuring Static Nodes

In some cases, the registration information is not accessible for a terminal or endpoint from any gatekeeper. This inaccessible registration information may be because the endpoint does not use RAS, is in an area where no gatekeeper exists, or is in a zone where the gatekeeper addressing is unavailable either through DNS or through configuration.

These endpoints can still be accessed via a gatekeeper by entering them as static nodes. To enter the endpoints as static nodes, obtain the address of the endpoint and then use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. For an explanation of the arguments, see Step 2 of the configuration task table in the "Starting a Gatekeeper" section on page 340.
Step 3	`Router(config-gk)# alias static` *ip-signalling-addr* [*port*] **gkid** *gatekeeper-name* [**ras** *ip-ras-addr port*] [**terminal** | **mcu** | **gateway** {**h320** |**h323-proxy** | **voip**}] [**e164** *e164-address*] [**h323id** *h323-id*]	Creates a static entry in the local alias table for each E.164 address. Repeat this step for each E.164 address you want to add for the endpoint. The keywords and arguments are as follows: • *ip-signalling-addr*—Specifies the IP address of the H.323 node, used as the address to signal when establishing a call. • *port*—(Optional) Specifies the port number other than the endpoint call-signaling well-known port number (1720). • **gkid** *gatekeeper-name*—Specifies the name of the local gatekeeper of whose zone this node is a member. • **ras** *ip-ras-addr*—(Optional) Specifies the node remote access server (RAS) signaling address. If omitted, the *ip-signalling-addr* parameter is used in conjunction with the RAS well-known port. • *port*—(Optional) Specifies a port number other than the RAS well-known port number (1719). • **terminal—**(Optional) Indicates that the alias refers to a terminal. • **mcu**—(Optional) Indicates that the alias refers to a multiple control unit (MCU). • **gateway**—(Optional) Indicates that the alias refers to a gateway. • **h320**—(Optional) Indicates that the alias refers to an H.320 node.**h320**—(Optional) Indicates that the alias refers to an H.320 node.

Command	Purpose
	• **h-323 proxy**—(Optional) Indicates that the alias refers to an H.323 proxy. • **voip**—(Optional) Indicates that the alias refers to VoIP. • **e164** *e164-address*—(Optional) Specifies the node E.164 address. This keyword and argument can be used more than once to specify as many E.164 addresses as needed. Note that there is a maximum number of 128 characters that can be entered for this address. To avoid exceeding this limit, you can enter multiple **alias static** commands with the same call-signaling address and different aliases. • **h323-id** *h323-id*—(Optional) Specifies the node H.323 alias. This keyword and argument can be used more than once to specify as many H.323 identification (ID) aliases as needed. Note that there is a maximum number of 256 characters that can be entered for this address. To avoid exceeding this limit, you can enter multiple commands with the same call signaling address and different aliases.

Configuring H.323 Users via RADIUS

To authenticate H.323 users via RADIUS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) access model.
Step 2	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Sets AAA authentication at login. The keywords and arguments are as follows: • **default**—Uses the listed authentication methods that follow this keyword as the default list of methods when a user logs in. • *list-name*—Specifies the character string used to name the list of authentication methods activated when a user logs in. • *method1* [*method2...*]—Specifies that at least one of the keywords described below be used: – **enable**—Uses the enable password for authentication. – **krb5**—Uses Kerberos 5 for authentication..

Command	Purpose
	– **krb5-telnet**—Uses the Kerberos 5 Telnet authentication protocol when using Telnet to connect to the router. – **line**—Uses the line password for authentication. – **local**—Uses the local username database for authentication – **local-case**—Uses case-sensitive local username authentication. – **none**—Uses no authentication. – **group radius**—Uses the list of all RADIUS servers for authentication. – **group tacacs+**—Uses the list of all TACACS+ servers for authentication. – **group group-name**—Uses a subset of RADIUS or TACACS+ servers for authentication as defined by the **group server radius** or **aaa group server tacacs+** command.
Step 3 `Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies the RADIUS server host. The keywords and arguments are as follows: • *hostname*—Specifies the Domain Name System (DNS) name of the RADIUS server host. • *ip-address*—Specifies the IP address of the RADIUS server host. • **auth-port**—(Optional) Specifies the User Datagram Protocol (UDP) destination port for authentication requests. • *port-number*—(Optional) Specifies the port number for authentication requests; the host is not used for authentication if set to 0. If unspecified, the port number defaults to 1645. • **acct-port**—(Optional) Specifies the UDP destination port for accounting requests. • *port-number*—(Optional) Specifies the port number for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646. • **acct-port**—(Optional) Specifies the UDP destination port for accounting requests. • *port-number*—(Optional) Specifies the port number for accounting requests; the host is not used for accounting if set to 0. If unspecified, the port number defaults to 1646.

Command	Purpose
	• **timeout**—(Optional) Specifies the time interval (in seconds) for which the router waits for the RADIUS server to reply before retransmitting. This setting overrides the global value of the **radius-server timeout** command. If no timeout value is specified, the global value is used. Enter a value in the range of from 1 to 1000. • *seconds*—(Optional) Specifies the timeout value. Enter a value in the range of from 1 to 1000. If no timeout value is specified, the global value is used. • **retransmit**—(Optional) Specifies the number of times a RADIUS request is resent to a server if that server is not responding or responding slowly. This setting overrides the global setting of the **radius-server retransmit** command. • *retries*—(Optional) Specifies the retransmit value. Enter a value in the range of from 1 to 100. If no retransmit value is specified, the global value is used. • **key**—(Optional) Specifies the authentication and encryption key used between the router and the RADIUS daemon running on this RADIUS server. This key overrides the global setting of the **radius-server key** command. If no key string is specified, the global value is used. The key is a text string that must match the encryption key used on the RADIUS server. Always configure the key as the last item in the **radius-server host** command syntax. This is because the leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in the key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key. • *string*—(Optional) Specifies the authentication and encryption key for all RADIUS communications between the router and the RADIUS server. This key must match the encryption used on the RADIUS daemon. All leading spaces are ignored, but spaces within and at the end of the key are used. If you use spaces in your key, do not enclose the key in quotation marks unless the quotation marks themselves are part of the key.

	Command	Purpose
Step 4	`Router(config)# radius-server key {0 string | 7 string | string}`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. The arguments are as follows: • **0**—Specifies that an unencrypted key will follow. • *string*—Specifies the unencrypted (cleartext) shared key. • **7**—Specifies that a hidden key will follow. • *string*—Specifies the hidden shared key. • *string*—Specifies the unencrypted (cleartext) shared key.
Step 5	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 6	`Router(config-gk)# security {any | h323-id | e164} {password default password | password separator character}`	Enables authentication and authorization on a gatekeeper. The keywords and arguments are as follows: • **any**—Uses the first alias of an incoming Registration, Admission, and Status (RAS) registration, regardless of its type, as the means of identifying the user to RADIUS/TACACS+. • **h323-id**—Uses the first H.323 ID type alias as the means of identifying the user to RADIUS/TACACS+. • **e164**—Uses the first E.164 address type alias as the means of identifying the user to RADIUS/TACACS+. • **password default** *password*—Specifies the default password that the gatekeeper associates with endpoints when authenticating them with an authentication server. The password must be identical to the password on the authentication server.

Command	Purpose
	• **password separator** *character*—Specifies the character that endpoints use to separate the H.323-ID from the piggybacked password in the registration. This allows each endpoint to supply a user-specific password. The separator character and password will be stripped from the string before it is treated as an H.323-ID alias to be registered. Note that passwords may be piggybacked only in the H.323-ID, not the E.164 address. This is because the E.164 address allows a limited set of mostly numeric characters. If the endpoint does not wish to register an H.323-ID, it can still supply an H.323-ID that consists of just the separator character and password. This will be understood to be a password mechanism, and no H.323-ID will be registered.

After the previous steps have been completed, enter each user into the RADIUS database using either the default password if using the **security password default** command or the actual passwords if using the piggybacked password mechanism as the RADIUS authentication for that user. Enter either the user H.323-ID or the E.164 address, depending on how the gatekeeper was configured.

For more information about configuring AAA services or RADIUS, refer to the *Cisco IOS Security Configuration Guide*.

Configuring a RADIUS/AAA Server

To configure the RADIUS/AAA server with information about the gatekeeper for your network installation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the authentication, authorization, and accounting (AAA) model.
Step 2	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Sets AAA authorization at login. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring H.323 Users via RADIUS" section on page 350.
Step 3	`Router(config)# radius-server deadtime minutes`	Improves the server response time when some servers might be unavailable. The *minutes* argument specifies the length of time, in minutes, for which a RADIUS server is skipped over by transaction requests, up to a maximum of 1440 minutes (24 hours).

	Command	Purpose
Step 4	`Router(config)# radius-server host {host-name | ip-address} [auth-port port-number] [acct-port port-number] [timeout seconds] [retransmit retries] [key string]`	Specifies the RADIUS server host. For an explanation of the keywords and arguments, see Step 3 in the configuration task table in the "Configuring H.323 Users via RADIUS" section on page 350.
Step 5	`Router(config)# radius-server key {0 string | 7 string | string}`	Sets the authentication and encryption key for all RADIUS communications between the router and the RADIUS daemon. For an explanation of the arguments, see Step 4 in the configuration task table in the "Configuring H.323 Users via RADIUS" section on page 350.

In addition to the above configuration, make sure that the following information is configured in your CiscoSecure AAA server:

- In the /etc/raddb/clients file, ensure that the following information is provided.

```
#Client Name            Key
#-----------       -------------------
gk215.cisco.com         testing123
```

Where:

gk215.cisco.com is resolved to the IP address of the gatekeeper requesting authentication.

- In the /etc/raddb/users file, ensure that the following information is provided:

```
taeduk@cisco.com  Password = "thiswouldbethepassword"
User-Service-Type = Framed-User,
Login-Service = Telnet
```

Where:

taeduk@cisco.com is the h323-id of the gateway authenticating to gatekeeper gk215.cisco.com.

Configuring User Accounting Activity for RADIUS

After AAA has been enabled and the gateway has been configured to recognize RADIUS as the remote security server providing authentication services, the next step is to configure the gateway to report user activity to the RADIUS server in the form of connection accounting records. To send connection accounting records to the RADIUS server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa accounting connection h323 {stop-only | start-stop | wait-start | none} [broadcast] group group-name`	Defines the accounting method list H.323 with RADIUS as a method. The keywords and arguments are as follows: • **stop-only**—Sends a "stop" accounting notice at the end of the requested user process. • **start-stop**—Sends a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process begins regardless of whether the "start" accounting notice was received by the accounting server. • **wait-start**—Sends a "start" accounting notice at the beginning of a process and a "stop" accounting notice at the end of a process. The "start" accounting record is sent in the background. The requested user process does not begin until the "start" accounting notice is received by the server. • **none**—Disables accounting services on this line or interface. • **broadcast**—(Optional) Enables sending accounting records to multiple AAA servers. Simultaneously sends accounting records to the first server in each group. If the first server is unavailable, failover occurs using the backup servers defined within that group. • **group** *group-name*—Specifies the server group to be used for accounting services. The following are valid server group names: – *string*—Specifies the character string used to name a server group. – **radius**—Uses list of all RADIUS hosts. – **tacacs+**—Uses list of all TACACS+ hosts.

	Command	Purpose
Step 2	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 3	`Router(config-gk)# aaa accounting`	Enables authentication, authorization, and accounting (AAA) of requested services for billing or security purposes when you use RADIUS or TACACS+.

For more information about AAA connection accounting services, refer to the *Cisco IOS Security Configuration Guide*.

Configuring E.164 Interzone Routing

With Cisco IOS Release 12.0(3)T and later releases, interzone routing may be configured using E.164 addresses. To configure interzone routing in the E.164 address space, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. For an explanation of the arguments, see Step 2 of the configuration task table in the "Starting a Gatekeeper" section on page 340.
Step 3	`Router(config-gk)# zone remote` *other-gatekeeper-name other-domain-name other-gatekeeper-ip-address* [*port-number*]	Statically specifies a remote zone if Domain Name System (DNS) is unavailable or undesirable. Enter this command for each gatekeeper. The arguments are as follows: • *other-gatekeeper-name*—Specifies the name of the remote gatekeeper. • *other-domain-name*—Specifies the domain name of the remote gatekeeper. • *other-gatekeeper-ip-address*—Specifies the IP address of the remote gatekeeper. • *port-number*—(Optional) Specifies the Registration, Admission, and Status (RAS) signaling port number for the remote zone. Value ranges are from 1 to 65,535. If this option is not set, the default is the well-known RAS port number 1719.

	Command	Purpose
Step 4	`Router(config-gk)# zone prefix gatekeeper-name e164-prefix [blast | seq] [gw-priority priority gw-alias [gw-alias, ...]]`	Adds a prefix to the gatekeeper zone list. For an explanation of the keywords and arguments, see Step 3 of the configuration task table in the "Starting a Gatekeeper" section on page 340.
Step 5	`Router(config-gk)# gw-type-prefix type-prefix [[hopoff gkid1] [hopoff gkid2] [hopoff gkidn] [seq | blast]] [default-technology] [[gw ipaddr ipaddr [port]]...]`	Configures the gatekeepers to service a technology zone and specifies whether location requests (LRQs) should be sent in blast or sequential fashion. The default is sequential. For an explanation of the keywords and arguments, see Step 2 of the configuration task table in the "Configuring Redundant Gatekeepers for a Technology Prefix" section on page 347.

Configuring H.323 Version 2 Features

To configure H.323 Version 2 features using the Cisco gatekeeper, perform the following configuration tasks. The first two tasks are required; the others are optional. Make sure that you include a priority value for selecting between multiple gateways when you configure the gatekeeper.

- Configuring a Dialing Prefix for Each Gateway, page 359 (Required)
- Configuring a Gatekeeper for Interaction with External Applications, page 361 (Required)
- Configuring a Prefix to a Gatekeeper Zone List, page 362 (Optional)
- Configuring Gatekeeper Triggers for Interaction with External Applications, page 363 (Optional)
- Configuring Inbound or Outbound Gatekeeper Proxied Access, page 366 (Optional)
- Configuring a Forced Disconnect on a Gatekeeper, page 368 (Optional)

See the "H.323 Applications" chapter for further information on H.323 Version 2 features supported by Cisco IOS software.

Configuring a Dialing Prefix for Each Gateway

To configure a dialing prefix for each gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone local` *gatekeeper-name domain-name* [*ras-IP-address*]	Specifies a zone controlled by a gatekeeper. The arguments are as follows: • *gatekeeper-name*—Specifies the gatekeeper name or zone name. This is usually the fully domain-qualified host name of the gatekeeper. For example, if the *domain-name* is cisco.com, the *gatekeeper-name* might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the *gatekeeper-name* for each zone should be some unique string that has a mnemonic value. • *domain-name*—Specifies the domain name served by this gatekeeper. • *ras-IP-address*—(Optional) The IP address of one of the interfaces on the gatekeeper. When the gatekeeper responds to gatekeeper discovery messages, it signals the endpoint or gateway to use this address in future communications. **Note** Setting this address for one local zone makes it the address used for all local zones.

Command	Purpose
Step 3 `Router(config-gk)# zone prefix gatekeeper-name e164-prefix [gw-priority pri-0-to-10 gw-alias [gw-alias, ...]]`	Adds a prefix to the gatekeeper zone list. To remove knowledge of a zone prefix, use the **no** form of this command with the gatekeeper name and prefix. To remove the priority assignment for a specific gateway, use the **no** form of this command with the **gw-priority** option. The keywords and arguments are as follows: • *gatekeeper-name*—Specifies the name of a local or remote gatekeeper, which must have been defined by using the **zone local** or **zone remote** command. • *e164-prefix*—Specifies an E.164 prefix in standard form followed by dots (.). Each dot represents a number in the E.164 address. For example, 212..... is matched by 212 and any seven numbers. **Note** Although a dot representing each digit in an E.164 address is the preferred configuration method, you can also enter an asterisk (*) to match any number of digits. • **gw-priority** *pri-0-to-10 gw-alias*—(Optional) Use the **gw-priority** option to define how the gatekeeper selects gateways in its local zone for calls to numbers that begin with prefix *e164-prefix*. Do not use this option to set priority levels for a prefix assigned to a remote gatekeeper. Use values from 0 to 10. A 0 value prevents the gatekeeper from using the gateway *gw-alias* for that prefix. Value 10 places the highest priority on gateway *gw-alias*. If you do not specify a priority value for a gateway, the value 5 is assigned. To assign the same priority value for one prefix to multiple gateways, list all the gateway names after the *pri-0-to-10* value. The *gw-alias* name is the H.323 identification (ID) of a gateway that is registered or that will register with the gatekeeper. This name is set on the gateway with the **h323-gateway voip h.323-id** command.

To put all your gateways in the same zone, use the **gw-priority** option and specify which gateways are used for calling different area codes. For example:

```
zone local localgk xyz.com
zone prefix localgk 408.......
zone prefix localgk 415....... gw-priority 10 gw1 gw2
zone prefix localgk 650....... gw-priority 0 gw1
```

The above commands accomplish the following:

- Domain xyz.com is assigned to gatekeeper localgk.
- Prefix 408 is assigned to gatekeeper localgk, and no gateway priorities are defined for it; therefore, all gateways registering to localgk can be used equally for calls to the 408 area code. No special gateway lists are built for the 408 prefix; a selection is made from the master list for the zone.
- The prefix 415 is added to gatekeeper localgk, and priority 10 is assigned to gateways gw1 and gw2.

- Prefix 650 is added to gatekeeper localgk, and priority 0 is assigned to gateway gw1.
- A priority 0 is assigned to gateway gw1 to exclude it from the gateway pool for prefix 650. When gw2 registers with gatekeeper localgk, it is added to the gateway pool for each prefix as follows:
  - For gateway pool for 415, gateway gw2 is set to priority 10.
  - For gateway pool for 650, gateway gw2 is set to priority 5.

Configuring a Gatekeeper for Interaction with External Applications

There are two ways of configuring the gatekeeper for interaction with an external application. You can configure a port number where the gatekeeper listens for dynamic registrations from applications. Using this method, the application connects to the gatekeeper and specifies the trigger conditions in which it is interested.

The second method involves using the command-line interface to statically configure the information about the application and its trigger conditions, in which case the gatekeeper initiates a connection to the external application.

To configure a gatekeeper (sj.xyz.com) that uses port 20000 for a specific connection with an external server (Server-123), use the following commands beginning in global configuration mode. Server-123 has a number of triggers that are used to maintain a database of active gateways, which are used for active call resolution.

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config)# server registration-port` *port-number*	Establishes the server registration port that is used for communication between the server and the gatekeeper. The *port-number* argument specifies a single range of values from 1 through 65,535 for the port number on which the gatekeeper listens for external server connections.

Server-123 establishes a connection with gatekeeper sj.xyz.com on port 20000 and sends a REGISTER RRQ message to gatekeeper sj.xyz.com to express interest in all RRQs from voice gateways that support a technology prefix of 1# or 2#.

The following is an example of a registration message:

```
REGISTER RRQ
Version-id:1
From:Server-123
To:sj.xyz.com
Priority:2
Notification-Only:
Content-Length:29

t=voice-gateway
p=1#
p=2#
```

When gatekeeper sj.xyz.com receives this message, the information supplied in the message is added to the trigger list. Then, when an endpoint registers with this gatekeeper by using an RRQ that matches the specified trigger condition in the message, the gatekeeper sends a notification to Server-123.

The following is an example of an RRQ notification sent from the gatekeeper to the server when the above trigger condition matches:

```
REQUEST RRQ
Version-id:1
From:sj.xyz.com
To:Server-123
Notification-Only:
Content-Length:89

c=I:172.18.00.00:1720
r=I:172.20.01.40:16523
a=H:gw3-sj
t=voice-gateway
p=1# 2#
```

Configuring a Prefix to a Gatekeeper Zone List

To add a prefix to a gatekeeper zone list, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# zone prefix` *gatekeeper-name e164-prefix* [**blast** | **seq**] [**gw-priority** *priority gw-alias* [*gw-alias, ...*]]	Adds a prefix to the gatekeeper zone list. For an explanation of the keywords and arguments, see Step 3 of the configuration task table in the "Starting a Gatekeeper" section on page 340.

Note Note that the **zone prefix** command matches a prefix to a gateway. It does not register the gateway. The gateway must register with the gatekeeper before calls can be completed through that gateway.

Verifying an Added Prefix

To view the prefixes added to the gatekeeper zone list, use the **show gatekeeper zone prefix** command. To see gatekeeper zone information, use the **show gatekeeper zone status** command.

Configuring Gatekeeper Triggers for Interaction with External Applications

To establish statically configured triggers on a router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config)# server trigger {arq | lcf | lrj | lrq | rrq | urq} gkid priority server-id server-ip-address server-port`	Configures a static server trigger for external applications. Enter the **all** form of the **no server trigger all** command to remove every static trigger that you configured if you want to delete them all. The keywords and arguments are as follows: • *all*—Deletes all command-line interface- (CLI-) configured triggers. • **arq, lcf, lrj, lrq, rrq, urq**—Specifies Registration, Admission, and Status (RAS) message types. Use these message types to specify a submode in the gatekeeper configuration mode where you configure a trigger for the gatekeeper to act upon. Specify only one message type per server trigger command. There is a different trigger submode for each message type. Each trigger submode has its own set of applicable commands. • *gkid*—Specifies the local gatekeeper identifier. • *priority*—Specifies the priority for each trigger. The range is from 1 through 20, with 1 being the highest priority. • *server-id*—Specifies the identification (ID) number of the external application. • *server-ip-address*—Specifies the IP address of the server. • *server-port*—Specifies the port on which the Cisco IOS gatekeeper listens for messages from the external server connection.
Step 3	`Router(config)# info-only`	Indicates to the Cisco IOS gatekeeper that messages that meet the specified trigger parameters should be sent as notifications only and that the Cisco IOS gatekeeper should not wait for a response from the external application.

	Command	Purpose
Step 4	`Router(config)# destination-info {e164 | email-id | h323-id} value`	Configures a trigger that is based on a particular destination. Repeat this command for more destinations. The keywords and arguments are as follows: • **e164**—Indicates that the destination address is an E.164 address. • **email-id**—Indicates that the destination address is an e-mail ID. • **h323-id**—Indicates that the destination address is an H.323 ID. • *value*—Specifies the value against which to compare the destination address in the RAS messages. For E.164 addresses, the following wildcards can be used: – A trailing series of periods, each of which represents a single character. – A trailing asterisk, which represents one or more characters.
Step 5	`Router(config)# redirect-reason value`	Configures a trigger that is based on a specific redirect reason. Repeat this command for more destinations. The *value* argument specifies the value against which to compare the redirect reason in the RAS messages. Possible values are from 0 to 65,535. Currently used redirect reasons are as follows: • 0—Unknown reason. • 1—Call forwarding is busy or called DTE is busy. • 2—Call forwarded; no reply. • 4—Call deflection. • 9—Called DTE out of order. • 10—Call forwarding by the call DTE 15—Call forwarding unconditionally. • 15—Call forwarding unconditionally.

	Command	Purpose
Step 6	`Router(config)# remote-ext-address [e164] value`	Limits the qualifying messages based on the remote extension address. Repeat this command for more destinations. The keywords and arguments are as follows: • **e164**—(Optional) Indicates that the remote extension address is an E.164 address. • *value*—Specifies the value against which to compare the destination address in the RAS messages. The following wildcards can be used: – A trailing series of periods, each of which represents a single character. – A trailing asterisk, which represents one or more characters.
Step 7	`Router(config)# endpoint-type value`	Configures a trigger that is based on a specific endpoint. Repeat this command for more destinations. The *value* argument specifies the value against which to compare the endpoint type in the RAS messages. The possible values are as follows: • **gatekeeper**—Specifies that the endpoint is an H.323 gatekeeper. • **h320-gateway**—Specifies that the endpoint is an H.320 gateway. • **mcu**—Specifies that the endpoint is a multipoint control unit (MCU). • **other-gateway**—Specifies that the endpoint is a type of gateway not specified on this list. • **proxy**—Specifies that the endpoint is an H.323 proxy. • **terminal**—Specifies that the endpoint is an H.323 terminal. • **voice-gateway**—Specifies that the endpoint is a voice type gateway.
Step 8	`Router(config)# supported-prefix value`	Configures a trigger that is based on a specific supported prefix. Repeat this command for more destinations. The *value* argument specifies the value against which to compare the supported prefix in the RAS messages. The possible values are any E.164 pattern used as a gateway technology prefix. The value string may contain any of the following: 0123456789#*,

Note Repeat Steps 2 through 8 in the above configuration task table for each trigger that you want to define.

Note To remove a trigger, enter the **no server trigger** command. To temporarily suspend a trigger, enter the trigger configuration mode, as described in Step 2, and enter the **shutdown** subcommand.

Configuring Inbound or Outbound Gatekeeper Proxied Access

By default, a gatekeeper will offer the IP address of the local proxy when queried by a remote gatekeeper (synonymous with remote zone). This is considered proxied access. Before Cisco IOS Release 12.0(5)T, the local gatekeeper was configured using the **zone access** command to offer the address of the local endpoint instead of the address of the local proxy (considered direct access).

Note The **use-proxy** command replaces the **zone access** command. The **use-proxy** command, configured on a local gatekeeper, affects only the use of proxies for incoming calls (that is, it does not affect the use of local proxies for outbound calls). When originating a call, a gatekeeper will use a proxy only if the remote gatekeeper offers a proxy at the remote end. A call between two endpoints in the same zone will always be a direct (nonproxied) call.

To configure a proxy for inbound calls from remote zones to gateways in its local zone and to configure a proxy for outbound calls from gateways in its local zone to remote zones, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# use-proxy` *local-zone-name* {**default** | **remote-zone** *remote-zone-name*} {**inbound-to** | **outbound-from**} {**gateway** | **terminal**}	Enables proxy communications for calls between local and remote zones. The keywords and arguments are as follows: • *local-zone-name*—Specifies the name or zone name of the gatekeeper, which is usually the fully domain-qualified host name of the gatekeeper. For example, if the domain name is cisco.com, the gatekeeper name might be gk1.cisco.com. However, if the gatekeeper is controlling multiple zones, the name of the gatekeeper for each zone should be a unique string that has a mnemonic value. • **default**—Defines the default proxy policy for all calls that are not defined by a **use-proxy** command that includes the **remote-zone** keyword. • **remote-zone** *remote-zone-name*—Defines a proxy policy for calls to or from a specific remote gatekeeper or zone.

Command	Purpose
	• **inbound-to**—Applies the proxy policy to calls that are inbound to the local zone from a remote zone. Each **use-proxy** command defines the policy for only one direction. • **outbound-from**—Applies the proxy policy to calls that are outbound from the local zone to a remote zone. Each **use-proxy** command defines the policy for only one direction. • **gateway**—Defines the type of local device to which the policy applies. The gateway option applies the policy only to local gateways. • **terminal**—Defines the type of local device to which the policy applies. The terminal option applies the policy only to local terminals.

Verifying Gatekeeper Proxied Access Configuration

Use the **show gatekeeper zone status** command to see information about the configured gatekeeper proxies and gatekeeper zone information (as shown in the following output).

Router# **show gatekeeper zone status**

```
                    GATEKEEPER ZONES
                    ================
GK name      Domain Name   RAS Address     PORT  FLAGS MAX-BW   CUR-BW
                                                     (kbps)    (kbps)
-------      -----------   -----------     ----  ----- ------    ------
sj.xyz.com   xyz.com       10.0.0.9 1719  LS             0
  SUBNET ATTRIBUTES :
    All Other Subnets :(Enabled)
  PROXY USAGE CONFIGURATION :
    inbound calls from germany.xyz.com :
      to terminals in local zone sj.xyz.com :use proxy
      to gateways in local zone sj.xyz.com  :do not use proxy
    outbound calls to germany.xyz.com
      from terminals in local zone germany.xyz.com :use proxy
      from gateways in local zone germany.xyz.com  :do not use proxy
    inbound calls from all other zones :
      to terminals in local zone sj.xyz.com :use proxy
      to gateways in local zone sj.xyz.com  :do not use proxy
    outbound calls to all other zones :
      from terminals in local zone sj.xyz.com :do not use proxy
      from gateways in local zone sj.xyz.com  :do not use proxy
tokyo.xyz.co xyz.com       172.21.139.89      1719  RS             0
milan.xyz.co xyz.com       172.16.00.00       1719  RS             0
```

Configuring a Forced Disconnect on a Gatekeeper

To force a disconnect on a gatekeeper, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear h323 gatekeeper call {all | local-callID local-callID}`	Forces a disconnect on a specific call or on all calls currently active on this gatekeeper. The keywords and arguments are as follows: • **all**—Forces all active calls currently associated with this gatekeeper to be disconnected. • **local-callID**—Forces a single active call associated with this gatekeeper to be disconnected. • *local-callID*—Specifies the local call identification number (CallID) that identifies the call to be disconnected.

To force a particular call to be disconnected (as opposed to all active calls on the H.323 gateway), use the local call identification number (CallID) to identify that specific call. Find the local CallID number for a specific call by using the **show gatekeeper calls** command; the ID number is displayed in the LocalCallID column.

Verifying a Forced Disconnect

To show the status of each ongoing call that a gatekeeper is aware of, use the **show gatekeeper calls** command. If you have forced a disconnect either for a particular call or for all calls associated with a particular H.323 gatekeeper, the system will not display information about those calls.

The following is sample output from the **show gatekeeper calls** command:

```
router# show gatekeeper calls

Total number of active calls =1
                    Gatekeeper Call Info
                    ====================
LocalCallID                     Age (secs)      BW
12-3339                         94              768 (Kbps)
 Endpt(s): Alias    E.164Addr    CallSignalAddr   Port   RASSignalAddr   Port
  src EP: epA                     10.0.0.11        1720   10.0.0.11        1700
  dst EP: epB2zoneB.com
  src PX: pxA                     10.0.0.1         1720   10.0.0.11        24999
  dst PX: pxB                     172.21.139.90    1720   172.21.139.90    24999
```

Configuring the Proxy

This section describes the following configuration tasks for configuring the proxy. Depending on your specific network design, either the first task or the second task is required.

- Configuring a Proxy Without ASR, page 369
- Configuring a Proxy with ASR, page 373

Configuring a Proxy Without ASR

To start the proxy without application-specific routing (ASR), start the proxy and then define the H.323 name, zone, and QoS parameters on the interface whose IP address the proxy will use. To start the proxy without ASR, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# proxy h323`	Starts the proxy feature.

Step 2	`Router(config)# interface type number [name-tag]`
	
	Cisco 4000 Series with Channelized T1 or E1 and the Cisco MC3810
	`Router(config)# interface serial number:channel-group`
	
	To configure a subinterface, use these forms of the interface global configuration command:
	
	Cisco 7200 Series
	`Router(config)# interface type slot/port-adapter/port.subinterface-number [multipoint | point-to-point]`
	
	Cisco 7200 Series and Cisco 7500 Series with a Packet over SONET Interface Processor
	`Router(config)# interface type slot/port`
	
	Cisco 7500 Series
	`Router(config)# interface type slot/port-adapter.subinterface-number [multipoint | point-to-point][ethernet | serial]`
	
	Cisco 7500 Series with Channelized T1 or E1
	`Router(config)# interface serial slot/port:channel-group`
	
	Cisco 7500 Series with Ports on VIP Cards
	`Router(config)# interface type slot/port-adapter/port [ethernet | serial]`
		Configures an interface type and enters interface configuration mode.
		
		The keywords and arguments are as follows:
		
		• *type*—Specifies the type of interface to be configured. (See Table 29 that follows this configuration task table.)
		• *number*—Specifies the port, connector, or interface card number. On a Cisco 4000 series router, specifies the network process monitor (NPM) number. The numbers are assigned at the factory at the time of installation or when added to a system, and they can be displayed with the **show interfaces** command.
		• *name-tag*—(Optional) Specifies the logic name to identify the server configuration so that multiple entries of server configuration can be entered. This optional argument is for use with the Redundant Link Manager (RLM) feature.
		• *slot*—Specifies the number of the slot being configured. Refer to the appropriate hardware manual for slot and port information.
		• *port*—Specifies the number of the port being configured. Refer to the appropriate hardware manual for slot and port information.
		• *port-adapter*—Specifies the number of the port adapter being configured. Refer to the appropriate hardware manual for information about port adapter compatibility.
		• **ethernet**—(Optional) Specifies an Ethernet IEEE 802.3 interface.
		• **serial**—(Optional) Specifies a serial interface.

	Command	Purpose
		• **:***channel-group*—Specifies a T1 channel group number in the range 0 to 23 defined with the **channel-group** controller configuration command. On a dual port card, it is possible to run channelized on one port and primary rate on the other port. Cisco MC3810 specifies the T1/E1 channel group number in the range 0 to 23 defined with the **channel-group** controller configuration command. • *.subinterface-number*—Specifies a subinterface number in the range of 1 to 4,294,967,293. The number that precedes the period (.) must match the number to which this subinterface belongs. • **multipoint** | **point-to-point**—(Optional) Specifies a multipoint or point-to-point subinterface. There is no default.
Step 3	`Router(config-if)# h323 interface [port-number]`	Selects an interface whose IP address will be used by the proxy to register with the gatekeeper. The *port-number* argument specifies the port number on which the proxy will listen for incoming call setup requests: • The *port-number* range is from 1 to 65,356. The default port number for the proxy is 11,720 in -isx- or -jsx- Cisco IOS images. • The default port number for the proxy is 1720 in -ix- Cisco IOS images, which do not contain the Voice over IP (VoIP) gateway.
Step 4	`Router(config-if)# h323 h323-id h323-id`	Configures the proxy name. (More than one name may be configured if necessary.) The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper.

	Command	Purpose
Step 5	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. The keywords and arguments are as follows: • **id** *gatekeeper-id*—(Optional) Specifies the gatekeeper name. Typically, this is a Domain Name System (DNS) name, but it can also be a raw IP address in dotted form. If this parameter is specified, gatekeepers that have either the default or the explicit flags set for the subnet of the proxy will respond. If this parameter is not specified, only those gatekeepers with the default subnet flag will respond. • **ipaddr** *ipaddr* [*port*]—If this parameter is specified, the gatekeeper discovery message will be unicast to this address and, optionally, to the port specified. • **multicast**—If this parameter is specified, the gatekeeper discovery message will be multicast to the well-known Registration, Admission, and Status (RAS) multicast address and port.
Step 6	`Router(config-if)# h323 qos {ip-precedence value | rsvp {controlled-load | guaranteed-qos}}`	Enables quality of service (QoS) on the proxy. The keywords and arguments are as follows: • *ip-precedence value*—Specifies that Realtime Transport Protocol (RTP) streams should set their IP precedence bits to the specified value. • **rsvp** [**controlled-load**]—Specifies controlled load class of service. • **rsvp** [**guaranteed-qos**]—Specifies guaranteed QoS class of service.
Step 7	`Router(config-if)# ip route-cache [cbus] same-interface [flow] distributed`	Controls the use of high-speed switching caches for IP routing. The keywords are as follows: • **cbus**—(Optional) Enables both autonomous switching and fast switching. • **same-interface**—Enables fast-switching packets to back out through the interface on which they arrived.

Command	Purpose
	• **flow**—(Optional) Enables the Route Switch Processor (RSP) to perform flow switching on the interface. • **distributed**—Enables Versatile Interface Processor (VIP) distributed switching on the interface. This feature can be enabled on Cisco 7500 series routers with RSP and VIP controllers. If both the **ip route-cache flow** command and the **ip route-cache distributed** command are configured, the VIP does distributed flow switching. If only the **ip route-cache distributed** command is configured, the VIP does distributed switching.

Table 29 lists interface types that may be used for the *type* argument in Step 2 of the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.

Table 29 Interface Type Keywords

Keyword	Interface Type
async	Port line used as an asynchronous interface.
atm	ATM interface.
bri	ISDN BRI. This interface configuration is propagated to each of the B channels. B channels cannot be individually configured. The interface must be configured with dial-on-demand commands for calls to be placed on that interface.
dialer	Dialer interface.
ethernet	Ethernet IEEE 802.3 interface.
fastethernet	100-Mbps Ethernet interface on the Cisco 4500, Cisco 4700, Cisco 7000, and Cisco 7500 series routers.
fddi	FDDI interface.
group-async	Master asynchronous interface.
hssi	High-Speed Serial Interface (HSSI).
lex	LAN Extender (LEX) interface.
loopback	Software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface-number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.
null	Null interface.
port-channel	Port channel interface.
pos	Packet OC-3 interface on the Packet over SONET Interface Processor.
serial	Serial interface.
switch	Switch interface.
tokenring	Token Ring interface.

Table 29 Interface Type Keywords (continued)

Keyword	Interface Type
tunnel	Tunnel interface; a virtual interface. The number is the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
vg-anylan	100VG-AnyLAN port adapter.

Configuring a Proxy with ASR

To enable ASR on the proxy, start the proxy and then define the H.323 name, zone, and QoS parameters on the loopback interface. Next, determine which interface will be used to route the H.323 traffic and configure ASR on it. The ASR interface and all other interfaces must be separated so that routing information never travels from one to the other. There are two different ways to separate the ASR interface and all other interfaces:

- Use one type of routing protocol on the ASR interface and another on all the non-ASR interfaces. Include the loopback subnet in both routing domains.
- Set up two different autonomous systems, one that contains the ASR network and the loopback network and another that contains the other non-ASR networks and loopback network.

To ensure that the ASR interface and all other interfaces never route packets between each other, configure an access control list. (The proxy traffic will be routed specially because it is always addressed to the loopback interface first and then translated by the proxy subsystem.)

To start the proxy with ASR enabled on the proxy using one type of routing protocol on the ASR interface and another on all of the non-ASR interfaces, and with the loopback subnet included in both routing domains, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# proxy h323`	Starts the proxy.
Step 2	`Router(config)# interface` *type number* [*name-tag*]	Enters loopback interface configuration mode. For an explanation of the arguments, see Step 2 in the "Configuring a Proxy Without ASR" configuration task table. To configure a proxy with ASR enabled on the proxy using one type of routing protocol, the *type* argument is **loopback**. The **loopback** type specifies the software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The *number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces that you can create.

	Command	Purpose
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface. The keywords and arguments are as follows: • *ip-address*—Specifies the IP address. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 4	`Router(config-if)# h323 interface [port-number]`	Signals the proxy that this interface IP address is the one to use. For an explanation of the arguments, see Step 3 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 5	`Router(config-if)# h323 h323-id h323-id`	Configures the proxy name. (More than one name can be configured if necessary.) The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper.
Step 6	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. For an explanation of the keywords and arguments, see Step 5 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 7	`Router(config-if)# h323 qos {ip-precedence value | rsvp {controlled-load | guaranteed-qos}}`	Enables quality of service (QoS) on the proxy. For an explanation of the keywords and arguments, see Step 6 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 8	`Router(config-if)# interface type number [name-tag]`	If ASR is to be used, enters the interface through which outbound H.323 traffic should be routed. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.

	Command	Purpose
Step 9	`Router(config-if)# h323 asr [bandwidth max-bandwidth]`	Enables ASR and specifies the maximum bandwidth for a proxy. The keywords and arguments are as follows: • **bandwidth** *max-bandwidth*—Specifies the maximum bandwidth on the interface. Value ranges are from 1 to 10,000,000 kbps. If you do not specify a value for the *max-bandwidth* argument, the value defaults to the bandwidth on the interface. If you specify the *max-bandwidth* value as a value greater than the interface bandwidth, the bandwidth will default to the interface bandwidth.
Step 10	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up the ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 11	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode for a non-ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 13	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up a non-ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 14	`Router(config-if)# exit`	Exits interface configuration mode.
Step 15	`Router(config)# router rip`	Configures the Routing Information Protocol (RIP) for a non-ASR interface.
Step 16	`Router(config)# network network-number`	Specifies a list of networks for the RIP routing process or a loopback interface in an Interior Gateway Routing Protocol (IGRP) domain. The *network-number* argument specifies the IP address of the directly connected networks.
Step 17	`Router(config)# router igrp autonomous-system`	Configures Interior IGRP for an ASR interface. The *autonomous-system* argument specifies the autonomous system number that identifies the routes to the other IGRP routers. It is also used to tag the routing information.
Step 18	`Router(config)# network network-number`	Specifies a list of networks for the Routing Information Protocol (RIP) routing process. The *network-number* argument should include an ASR interface in an IGRP domain.
Step 19	`Router(config)# network loopback-addr`	Includes a loopback interface in an IGRP domain.

Command	Purpose
Step 20 `Router(config)# access-list access-list-number {permit | deny} source source-mask [destination destination-mask] {eq | neq} [[source-object] [destination-object] [identification] any]`	Creates an access list. The keywords and arguments are as follows: • *access-list-number*—Specifies the integer that you choose. The number should be between 300 and 399, and it uniquely identifies the access list. • **permit**—Permits access when there is an address match. • **deny**—Denies access when there is an address match. • *source*—Specifies the source address. DECnet addresses are written in the form area.node. For example, 50.4 is node 4 in area 50. All addresses are in decimal. • *source-mask*—Specifies the mask to be applied to the address of the source node. All masks are in decimal. • *destination*—(Optional) Specifies the DECnet address of the destination node in decimal format. DECnet addresses are written in the form area.node. For example, 50.4 is node 4 in area 50. All addresses are in decimal. • *destination-mask*—(Optional) Specifies the destination mask. DECnet addresses are written in the form area.node. For example, 50.4 is node 4 in area 50. All masks are in decimal. • **eq**—Specifies that the item matches the packet if all the specified parts of the source object, destination object, and identification match the data in the packet. • **neq**—Specifies that the item matches the packet if any of the specified parts do not match the corresponding entry in the packet. • *source-object*—(Optional) Contains the mandatory keyword **src** and one of the following optional keywords: – **eq | neq | lt | gt**—Specifies equal to, not equal to, less than, or greater than. These keywords must be followed by the argument *object-number*, a numeric DECnet object number. – **exp**—Stands for expression; followed by a regular-expression that matches a string. See the "Regular Expressions" appendix in the *Cisco IOS Dial Technologies Command Reference* for a description of regular expressions.

Command	Purpose
	• *destination-object*—(Optional) Contains the mandatory keyword **dst** and one of the following optional keywords: – **eq | neq | lt | gt**—Specifies equal to, not equal to, less than, or greater than. These keywords must be followed by the argument *object-number*, a numeric DECnet object number. – **exp**—Stands for expression; followed by a regular expression that matches a string. See the "Regular Expressions" appendix in the *Cisco IOS Dial Technologies Command Reference* for a description of regular expressions. – **uic**—Stands for user identification code; followed by a numeric UID expression. The argument [*group*, *user*] is a numeric UID expression. In this case, the bracket symbols are literal; they must be entered. The *group* and *user* parts can be specified either in decimal, in octal by prefixing the number with a 0, or in hex by prefixing the number with 0x. The **uic** expression displays as an octal number. • *identification*—(Optional) Uses any of the following three keywords: – **id**—Specifies regular expression; refers to the user ID. – **password**—Specifies regular expression; the password to the account. – **account**—Specifies regular expression; the account string. – **any**—(Optional) Specifies that the item matches if *any* of the specified parts *do* match the corresponding entries for *source-object*, *destination-object*, or *identification*.

	Command	Purpose
Step 21	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode on an ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 22	`Router(config-if)# ip access-group {access-list-number | access-list-name}{in | out}`	Controls access to an interface. Use this command to set the outbound access group and then the inbound access group. The keywords and arguments are as follows: • *access-list-number*—Specifies the number of an access list. This is a decimal number from 1 to 199 or from 1300 to 2699. • *access-list-name*—Name of an IP access list as specified by an IP **access-list** command. • **in**—Filters on inbound packets. • **out**—Filters on outbound packets.

Note ASR is not supported on Frame Relay or ATM interfaces for the Cisco MC3810 platform.

To start the proxy with ASR enabled on the proxy using two different autonomous systems (one that contains the ASR network and the loopback network and another that contains the other non-ASR networks and the loopback network), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# proxy h323`	Starts the proxy.
Step 2	`Router(config)# interface type number [name-tag]`	Enters loopback interface configuration mode. For an explanation of the arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369. To start the proxy with ASR enabled on the proxy using two different autonomous systems, the *type* argument is **loopback**. The **loopback** type specifies the software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The *number* argument is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.

	Command	Purpose
Step 3	`Router(config-if)# ip address ip-address mask [secondary]`	Sets a primary or secondary IP address for an interface. The keywords and arguments are as follows: • *ip-address*—Specifies the IP address. • *mask*—Specifies the mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 4	`Router(config-if)# h323 interface [port-number]`	Signals the proxy that this interface IP address is the one to use. For an explanation of the arguments, see Step 3 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 5	`Router(config-if)# h323 h323-id h323-id`	Configures the proxy name. (More than one name can be configured if necessary.) The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper.
Step 6	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. For an explanation of the keywords and arguments, see Step 5 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 7	`Router(config-if)# h323 qos {ip-precedence value | rsvp {controlled-load | guaranteed-qos}}`	Enables quality of service (QoS) on the proxy. The keywords and arguments are as follows: • **ip-precedence** *value*—Specifies that Real-time Transport Protocol (RTP) streams should set their IP precedence bits to the specified value. • **rsvp** {**controlled-load**}—Specifies controlled load class of service. • **rsvp** {**guaranteed-qos**}—Specifies guaranteed QoS class of service.
Step 8	`Router(config-if)# interface type number [name-tag]`	If application-specific routing (ASR) is to be used, enters the interface through which outbound H.323 traffic should be routed. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.

	Command	Purpose
Step 9	`Router(config-if)# h323 asr [bandwidth max-bandwidth]`	Enables ASR and specifies the maximum bandwidth for a proxy. The optional *max-bandwidth* argument specifies the maximum bandwidth on the interface. Value ranges are from 1 to 10,000,000 kbps. If you do not specify *max-bandwidth*, this value defaults to the bandwidth on the interface. If you specify *max-bandwidth* as a value greater than the interface bandwidth, the bandwidth will default to the interface bandwidth.
Step 10	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up the ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 11	`Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.
Step 12	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode on a non-ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 13	`Router(config-if)# ip address ip-address mask [secondary]`	Sets up a non-ASR interface network number. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 14	`Router(config-if)# exit`	Exits interface configuration mode.
Step 15	`Router(config)# router igrp autonomous-system`	Configures Interior Gateway Routing Protocol (IGRP) for a non-ASR interface. The *autonomous-system* argument specifies the autonomous system number that identifies the routes to the other IGRP routers. It is also used to tag the routing information.
Step 16	`Router(config)# network network-number`	Includes a non-ASR interface in an IGRP domain. The *network-number* argument specifies the IP address of the network of the directly connected networks.
Step 17	`Router(config)# network network-number`	Includes a loopback interface in an IGRP domain. The *network-number* argument specifies the IP address of the network of the directly connected networks.
Step 18	`Router(config)# router igrp autonomous-system`	Configures IGRP for an ASR interface. The *autonomous-system* argument specifies the autonomous system number that identifies the routes to the other IGRP routers. It is also used to tag the routing information.

	Command	Purpose
Step 19	`Router(config)# network network-number`	Specifies a list of networks for the Routing Information Protocol (RIP) routing process. The *network-number* argument should include an ASR interface in an IGRP domain.
Step 20	`Router(config)# network network-number`	Specifies a list of networks for the RIP routing process. The *network-number* argument should include a loopback interface in an IGRP domain.
Step 21	`Router(config)# access-list access-list-number {permit | deny} source source-mask [destination destination-mask] {eq | neq} [[source-object] [destination-object] [identification] any]`	Creates an access list. For an explanation of the keywords and arguments, see Step 20 in the configuration task table in the "Configuring a Proxy with ASR" section on page 373.
Step 22	`Router(config)# interface type number [name-tag]`	Enters interface configuration mode on an ASR interface. For an explanation of the keywords and arguments, see Step 2 in the configuration task table in the "Configuring a Proxy Without ASR" section on page 369.
Step 23	`Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out}`	Controls access to an interface. Use this command to set the outbound access group and then the inbound access group. The keywords and arguments are as follows: • *access-list-number*—Specifies the number of an access list. This is a decimal number from 1 to 199 or from 1300 to 2699. • *access-list-name*—Name of an IP access list as specified by an IP **access-list** command. • **in**—Filters on inbound packets. • **out**—Filters on outbound packets.

H.323 Gatekeeper Configuration Examples

This section includes the following configuration examples:

Configuring a Gatekeeper Example

The following is an annotated example of how to configure a gatekeeper:

```
hostname gk-eng.xyz.com
! This router serves as the gatekeeper for the engineering community.
! at xyz.com.
ip domain-name xyz.com
! Domain name of this company.
interface Ethernet0
 ip address 172.21.127.27 255.255.255.0
! This gatekeeper can be found at address 172.21.127.27.
gatekeeper
! Enter gatekeeper config mode.
 zone local gk-eng.xyz.com xyz.com
! Because a zone is, by definition, the area of control of a gatekeeper,
! we tend to use the terms "zone name" and "gatekeeper name" synonymously.
! Here we use the host name as the name of the gatekeeper and zone.
! This is not necessary, but it does simplify administration.
 zone remote gk-mfg.xyz.com xyz.com 172.12.10.14 1719
 zone remote gk-corp.xyz.com xyz.com 172.12.32.80 1719
! A couple of other zones within xyz.com. We make lots of calls
! between these departments, so we just configure these so we save
! a little time bypassing DNS lookup to find their gatekeepers.
 use-proxy gk-eng.xyz.com remote-zone gk-mfg.xyz.com direct
 use-proxy gk-eng.xyz.com remote-zone gk-corp.xyz.com direct
 use-proxy gk-eng.xyz.com default proxied
! We have good QoS on our local network, so we don't need proxies when
! calling between the xyz.com zones. But for all other zones, we want
! to use proxies.
zone subnet gk-eng.xyz.com 172.21.127.0/24 enable
no zone subnet gk-eng.xyz.com default enable
! We will accept registrations from our local subnet as long as they
! do not specify some other gatekeeper name. We will not accept any
! registrations from any other subnet.
 zone bw gk-eng.xyz.com 2000
```

```
! Preserve our good QoS by not allowing excessive amounts of H.323 traffic
! on the local network. This restricts the traffic within our zone,
! for both intra-zone and interzone calls, to 2 kbps at any given time.
 alias static 172.21.127.49 gkid gk-eng.xyz.com terminal h323id joeblow ras
172.21.127.49 1719
! The "user" has an H.323 terminal, which does not support RAS. So we have
! to configure his alias manually so that callers can find him.
```

Redundant Gatekeepers for a Zone Prefix Example

In the following example, two remote gatekeepers are configured to service the same zone prefix:

```
gatekeeper
 zone remote c2600-1-gk cisco.com 172.18.194.70 1719
 zone remote c2514-1-gk cisco.com 172.18.194.71 1719
 zone prefix c2600-1-gk 919.......
 zone prefix c2514-1-gk 919.......
```

Redundant Gatekeepers for a Technology Prefix Example

In the following example, two remote gatekeepers are configured to service the same technology prefix:

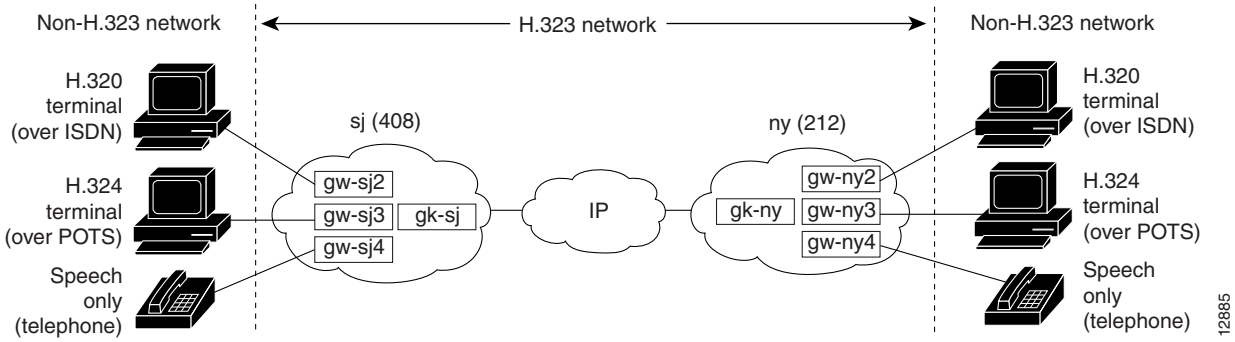
```
gatekeeper
 zone remote c2600-1-gk cisco.com 172.18.194.70 1719
 zone remote c2514-1-gk cisco.com 172.18.194.71 1719
 gw-type-prefix 3#* hopoff c2600-1-gk hopoff c2514-1-gk
```

E.164 Interzone Routing Example

Interzone routing may be configured by using E.164 addresses.

In this example, there are two gatekeepers that need to be able to resolve E.164 addresses. One is in San Jose and the other is in New York. (See Figure 78.)

Figure 78 E.164 Interzone Routing



In sj (San Jose in the 408 area code), the gateways are configured to register with gk-sj as follows:

- gw-sj2 configured to register with technology prefix 2#
- gw-sj3 configured to register with technology prefix 3#
- gw-sj4 configured to register with technology prefix 4#

Similarly, in ny (New York in the 212 area code), gateways are configured to register with gk-ny as follows:

- gw-ny2 configured to register with technology prefix 2#
- gw-ny3 configured to register with technology prefix 3#
- gw-ny4 configured to register with technology prefix 4#

For the gatekeeper for San Jose, the configuration commands are as follows:

```
gatekeeper
 zone local gk-sj cisco.com
 zone remote gk-ny cisco.com 172.21.127.27
 use-proxy gk-sj default direct
 zone prefix gk-sj 408.......
 zone prefix gk-ny 212.......
 gw-type-prefix 3# hopoff gk-sj
 gw-type-prefix 4# default-technology
```

For the gatekeeper for New York, the configuration commands are as follows:

```
gatekeeper
 zone local gk-ny cisco.com
 zone remote gk-sj cisco.com 172.21.1.48
 use-proxy gk-ny default direct
 zone prefix gk-sj 408.......
 zone prefix gk-ny 212.......
 gw-type-prefix 3# hopoff gk-ny
 gw-type-prefix 4# default-technology
```

When a call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
2#2125551212
```

Gatekeeper gk-sj recognizes that 2# is a technology prefix. It was not configured as such, but because gw-sj2 registered with it, the gatekeeper now treats 2# as a technology prefix. It strips the prefix, which leaves the telephone number 2125551212. This is matched against the zone prefixes that have been configured. It is a match for 212......., so gk-sj knows that gk-ny handles this call. Gatekeeper gk-sj forwards the entire address 2#2125551212 over to Gatekeeper gk-ny, which also looks at the technology prefix 2# and routes it to gw-ny2.

When a call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
2125551212
```

Gatekeeper gk-sj checks it against known technology prefixes but finds no match. It then checks it against zone prefixes and matches on 212..... for gk-ny, and therefore routes this call to gk-ny. Gatekeeper gk-ny does not have any local registrations for this address, and there is no technology prefix on the address, but the default prefix is 4#, and gw-ny4 is registered with 4#, so the call gets routed to gw-ny4.

Another call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
3#2125551212
```

The call has technology prefix 3#, which is defined as a local hopoff prefix, so gk-sj routes this call to gw-sj3, despite the fact that it has a New York zone prefix.

In this last example, a call is presented to gatekeeper gk-sj with the following target address in San Jose:

```
6505551212
```

Gatekeeper gk-sj checks for a technology prefix match but does not find one. It then searches for a zone prefix match and fails again. But there is a match for default gateway prefix of 4#, and gw-sj4 is registered with 4#, so the call is routed out on gw-sj4.

Configuring HSRP on the Gatekeeper Example

This sample configuration uses Ethernet 0 as the HSRP interface on both gatekeepers.

On the primary gatekeeper, enter these commands:

```
configure terminal
 ! Enter global configuration mode.
interface ethernet 0
 ! enter interface configuration mode for interface ethernet 0.
 standby 1 ip 172.21.127.55
 ! Member of standby group 1, sharing virtual address 172.21.127.55.
 standby 1 preempt
 ! Claim active role when it has higher priority.
 standby 1 timers 5 15
 ! Hello timer is 5 seconds; hold timer is 15 seconds.
 standby 1 priority 110
 ! Priority is 110.
```

On the backup gatekeeper, enter these commands:

```
configure terminal
interface ethernet 0
 standby 1 ip 172.21.127.55
 standby 1 preempt
 standby 1 timers 5 15
```

The configurations are identical except that gk2 has no standby priority configuration, so it assumes the default priority of 100—meaning that gk1 has a higher priority.

On both gk1 and gk2, set up identical gatekeeper mode configurations, as follows:

```
configure terminal
 ! Enter global configuration mode.
gatekeeper
 ! Enter gatekeeper configuration mode.
 zone local gk-sj cisco.com 172.21.127.55
 ! Define local zone using HSRP virtual address as gatekeeper RAS address.
.
.
.
 ! Various other gk-mode configurations.
 no shut
 ! Bring up the gatekeeper.

configure terminal
 ! Enter global configuration mode.
gatekeeper
 ! Enter gatekeeper configuration mode.
 zone local gk-sj cisco.com 172.21.127.55
 ! Define local zone using HSRP virtual address as gatekeeper RAS address.
 ! Note this uses the same gkname and address as on gk1.
.
.
! Various other gk-mode configurations.
 no shut
 ! Bring up the gatekeeper.
```

Note The **no shut** command is issued on both gatekeepers, primary and secondary. If the **show gatekeeper status** command is issued on the two gatekeepers, gk1 will show the following:

```
Gatekeeper State: UP
But gk2 will show the following:
Gatekeeper State: HSRP STANDBY
```

Using ASR for a Separate Multimedia Backbone Example

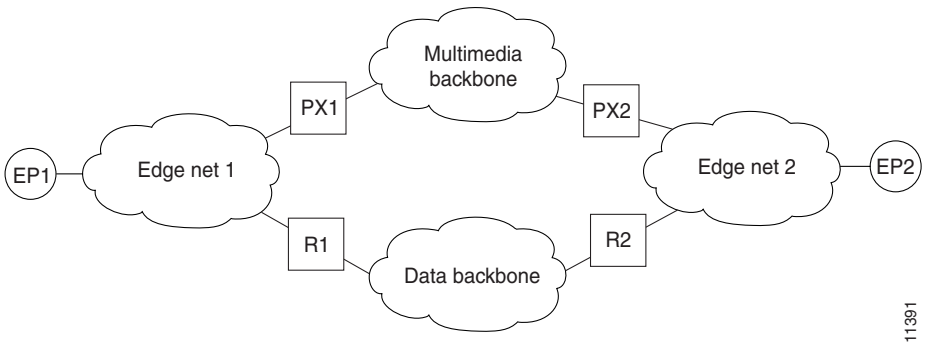
The examples in this section illustrate a separate multimedia backbone network dedicated to transporting only H.323 traffic. The closed functionality of the H.323 proxy is necessary for creating this type of backbone. Place a closed H.323 proxy on each edge of the multimedia backbone to achieve the following goals:

- The proxy directs all inter-proxy H.323 traffic, including Q.931 signaling, H.245, and media stream, to the multimedia backbone.
- The proxy shields the multimedia backbone so that routers on edge networks and other backbone networks are not aware of its existence. In this way, only H.323-compliant packets can access or traverse the multimedia backbone.
- The proxy drops any unintended non-H.323 packets that attempt to access the multimedia backbone.

This section contains the following subsections:

Figure 79 illustrates a network that has a multimedia backbone. A gatekeeper (not shown) in the edge network (zone) directs all out-of-zone H.323 calls to the closed proxy on the edge of that network. The closed proxy forwards this traffic to the remote zone through the multimedia backbone. A closed proxy and the edge router may reside in the same Cisco router, or they may be in separate routers, as shown in Figure 79.

Figure 79 Sample Network with Multimedia Backbone



Enabling the Proxy to Forward H.323 Packets

To enable the proxy to forward H.323 packets received from the edge network to the multimedia backbone, designate the interface that connects the proxy to the multimedia backbone to the ASR interface by entering the **h323 asr** command in interface configuration mode. Enabling the proxy to forward H.323 packets satisfies the first goal identified earlier in this section.

Because the proxy terminates two call legs of an H.323 call and bridges them, any H.323 packet that traverses the proxy will have the proxy address either in its source field or in its destination field.

To prevent problems that can occur in proxies that have multiple IP addresses, designate only one interface to be the proxy interface by entering the **h323 interface** command in interface configuration mode. Then all H.323 packets that originate from the proxy will have the address of this interface in their source fields, and all packets that are destined to the proxy will have the address of this interface in their destination fields.

Figure 79 illustrates that all physical proxy interfaces belong either to the multimedia network or to the edge network. These two networks must be isolated from each other for the proxy to be closed; however, the proxy interface must be addressable from both the edge network and the multimedia network. For this reason, a loopback interface must be created on the proxy and configured to the proxy interface.

It is possible to make the loopback interface addressable from both the edge network and the multimedia network without exposing any physical subnets on one network to routers on the other network. Only packets that originate from the proxy or packets that are destined to the proxy can pass through the proxy interface to the multimedia backbone in either direction. All other packets are considered unintended packets and are dropped. This can be achieved by configuring access control lists (ACLs) so that the closed proxy acts like a firewall that only allows H.323 packets to pass through the ASR interface. This satisfies the second goal identified earlier in this section, which is to ensure that only H.323-compliant packets can access or traverse the multimedia backbone.

Isolating the Multimedia Network

The last step is to configure the network so that non-H.323 traffic never attempts to traverse the multimedia backbone and so that it never risks being dropped by the proxy. This is achieved by completely isolating the multimedia network from all edge networks and from the data backbone and by configuring routing protocols on the various components of the networks.

The example provided in Figure 79 requires availability of six IP address classes, one for each of the four autonomous systems and one for each of the two loopback interfaces. Any Cisco-supported routing protocol can be used on any of the autonomous systems, with one exception: Routing Information Protocol (RIP) cannot be configured on two adjacent autonomous systems because this protocol does not include the concept of an autonomous system. The result would be the merging of the two autonomous systems into one.

If the number of IP addresses are scarce, use subnetting, but the configuration can get complicated. In this case, only the Enhanced IGRP, Open Shortest Path First (OSPF), and RIP Version 2 routing protocols, which allow variable-length subnet masks (VLSMs), can be used.

Assuming these requirements are met, configure the network illustrated in Figure 79 as follows:

- Configure each of the four networks as a separate routing autonomous system and do not redistribute routes between the multimedia backbone and any other autonomous system.
- Create a loopback interface on the proxy and configure it to be the proxy interface. That way no subnets of the multimedia backbone will be exposed to the edge network, or the other way around.
- To ensure that the address of the loopback interface does not travel outside the edge network, configure the appropriate distribution list on the edge router that connects the edge network to the data backbone. Configuring the appropriate distribution list guarantees that any ongoing H.323 call will be interrupted if the multimedia backbone fails. Otherwise, H.323 packets that originate from one proxy and that are destined to another proxy might discover an alternate route using the edge networks and the data backbone.

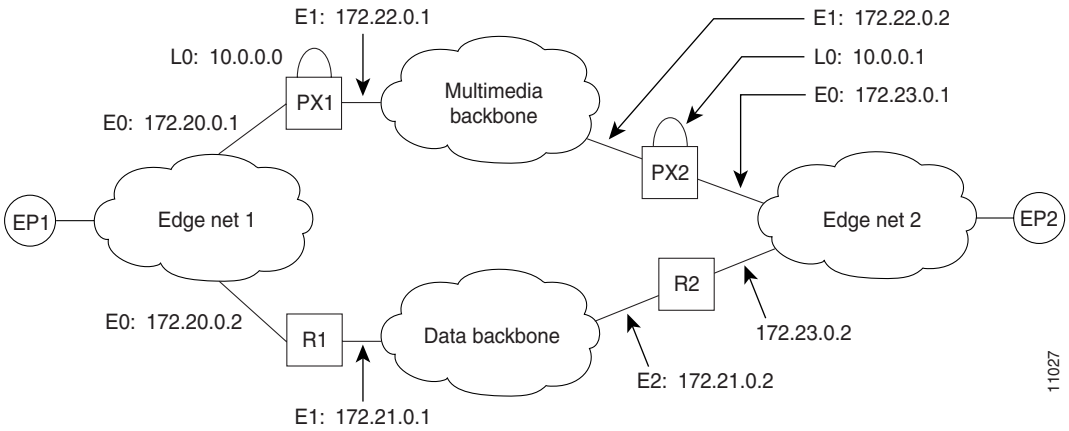
In some topologies, the two edge networks and the data backbone may be configured as a single autonomous system, but it is preferable to separate them as previously described because they are different networks with different characteristics.

The following examples illustrate the router configuration that is relevant to the closed proxy operation.

Configuring a Co-Edge Proxy with ASR Without Subnetting Example

See Figure 80 and the following configuration examples to see how to configure RIP on the two edge networks and how to configure IGRP on the two backbone networks.

Figure 80 Sample Configuration Without Subnetting



PX1 Configuration

The following output is for the PX1 configuration:

```
!
proxy h323
!
interface Loopback0

 ip address 10.0.0.0 255.0.0.0
!Assume PX1 is in Zone 1, and the gatekeeper resides in the same routers as PX1:
 h323 interface
 h323 h323-id PX1@zone1.com
 h323 gatekeeper ipaddr 10.0.0.0
!
```

```
interface Ethernet0
 ip address 172.20.0.1 255.255.0.0
!
interface Ethernet1
 ip address 172.22.0.1 255.255.0.0
 ip access-group 101 in
 ip access-group 101 out
 h323 asr
!
router rip
 network 172.20.0.0
 network 10.0.0.0
!
router igrp 4000
 network 172.22.0.0
 network 10.0.0.0
!
access-list 101 permit ip any host 10.0.0.0
access-list 101 permit ip host 10.0.0.0 any
access-list 101 permit igrp any any
```

R1 Configuration

The following output is for the R1 configuration:

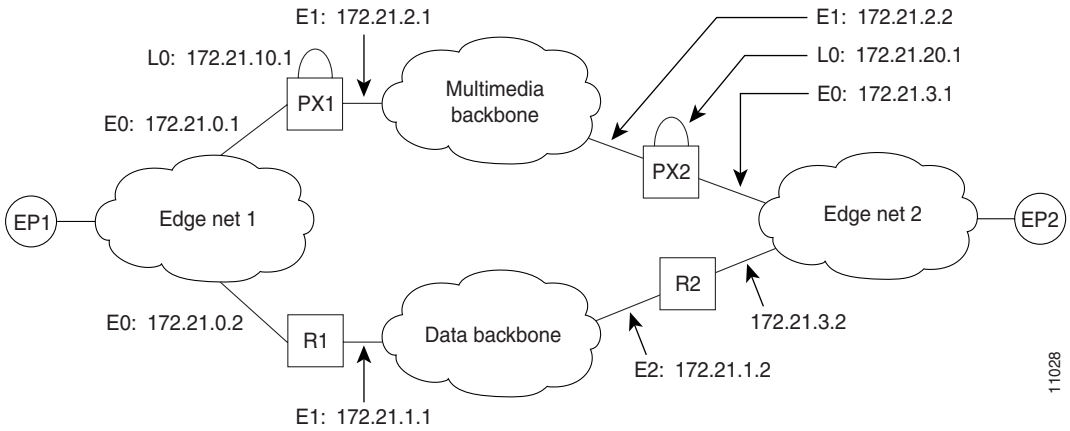
```
!
interface Ethernet0
 ip address 172.20.0.2 255.255.0.0
!
interface Ethernet1
 ip address 172.21.0.1 255.255.0.0
!
router rip
 redistribute igrp 5000 metric 1
 network 172.20.0.0
!
router igrp 5000
 redistribute rip metric 10000 10 255 255 65535
 network 172.21.0.0
 distribute-list 10 out
!
access-list 10 deny ip 10.0.0.0 255.255.255
access-list 10 permit any
```

Note The configuration for PX2 and R2 is the same as that for PX1 and R1.

Co-Edge Proxy with Subnetting Example

Figure 81 and the examples that follow illustrate how to configure Enhanced IGRP on all networks.

Figure 81 Sample Configuration with Subnetting



PX1 Configuration

The following output is for the PX1 configuration:

```
!
proxy h323
!
interface Loopback0
 ip address 172.21.10.1 255.255.255.192
 h323 interface
 h323 h323-id PX1@zone1.com
 h323 gatekeeper ipaddr 172.21.20.1
!
interface Ethernet0
 ip address 172.21.0.1 255.255.255.192
!
interface Ethernet1
 ip address 172.21.2.1 255.255.255.192
 ip access-group 101 in
 ip access-group 101 out
 h323 asr
!
router eigrp 4000
 redistribute connected metric 10000 10 255 255 65535
 passive-interface Ethernet1
 network 172.21.0.0
 distribute-list 10 out
 no auto-summary
!
router eigrp 5000
 redistribute connected metric 10000 10 255 255 65535
 passive-interface Ethernet0
 network 172.21.0.0
 distribute-list 11 out
 no auto-summary
!
access-list 10 deny 172.21.2.0 0.0.0.63
access-list 10 permit any
```

```
access-list 11 deny 172.21.0.0 0.0.0.63
access-list 11 permit any
access-list 101 permit ip any host 172.21.10.1
access-list 101 permit ip host 172.21.10.1 any
access-list 101 permit eigrp any any
```

R1 Configuration

The following output is for the R1 configuration:

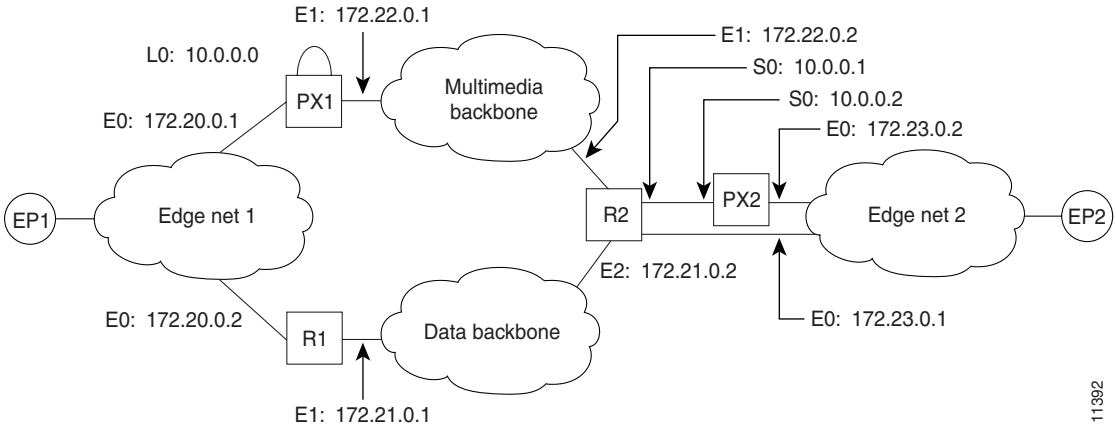
```
!
interface Ethernet0
 ip address 172.21.0.2 255.255.255.192
!
interface Ethernet1
 ip address 172.21.1.1 255.255.255.192
!
router eigrp 4000
 redistribute eigrp 6000 metric 10000 10 255 255 65535
 passive-interface Ethernet1
 network 172.21.0.0
 no auto-summary
!
router eigrp 6000
 redistribute eigrp 4000 metric 10000 10 255 255 65535
 passive-interface Ethernet0
 network 172.21.0.0
 distribute-list 10 out
 no auto-summary
!
access-list 10 deny 172.21.10.0 0.0.0.63
access-list 10 permit any
```

Note The configuration for PX2 and R2 is the same as that for PX1 and R1.

Configuring an Inside-Edge Proxy with ASR Without Subnetting Example

The configuration of the co-edge proxy in Edge net 1 has already been presented above. Figure 82 illustrates the configuration of the inside-edge proxy PX2 and edge router R2 of Edge net 2. RIP is used on the edge networks. IGRP is used on the data backbone and the multimedia backbone.

Figure 82 Edge Net 2 with Inside-Edge Proxy and No Subnetting



PX2 Configuration

The following output is for the PX2 configuration:

```
!
proxy h323
!
interface Ethernet0
 ip address 172.23.0.2 255.255.0.0
!
interface Serial0
 ip address 10.0.0.2 255.0.0.0
 ip access-group 101 in
 ip access-group 101 out
 h323 interface
 h323 asr
 h323 h323-id PX2@zone2.com
 h323 gatekeeper ipaddr 10.0.0.2
!
router rip
 redistribute connected metric 10000 10 255 255 65535
 network 172.23.0.0
!
access-list 101 permit ip any host 10.0.0.2
access-list 101 permit ip host 10.0.0.2 any
```

R2 Configuration

The following output is for the R2 configuration:

```
!
interface Ethernet0
 ip address 172.23.0.1 255.255.0.0
!
interface Ethernet1
 ip address 172.22.0.1 255.255.0.0
```

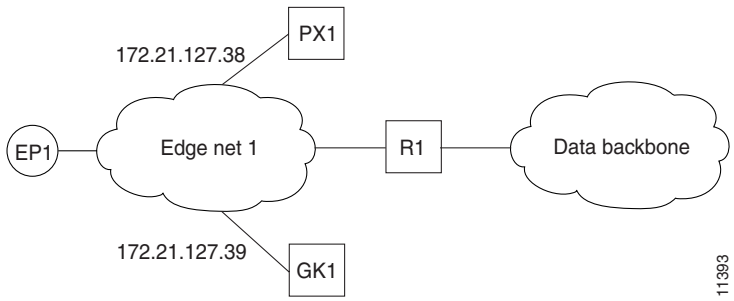
```
 ip access-group 101 in
 ip access-group 101 out
!
interface Ethernet2
 ip address 172.21.0.2 255.255.0.0
!
interface Serial0
 ip address 10.0.0.1 255.0.0.0
!
router rip
 redistribute igrp 5000 metric 1
 network 172.23.0.0
!
router igrp 4000
 network 10.0.0.0
 network 172.22.0.0
!
router igrp 5000
 redistribute rip metric 10000 10 255 255 65535
 network 172.21.0.0
 distribute-list 10 out
!
ip route 10.0.0.2 255.255.255.255 Serial0
access-list 10 deny ip 10.0.0.0 255.255.255
access-list 10 permit any
access-list 101 permit ip any host 10.0.0.2
access-list 101 permit ip host 10.0.0.2 any
```

Note To guarantee that all traffic between the proxy and other proxies is carried over the multimedia backbone, run IGRP 4000 on the 10.0.0.0 network and on the 172.22.0.0 network. Make sure that the H.323 proxy interface address (10.0.0.2) is not advertised over the data network (distribution list 10 in IGRP 5000). Doing this also eliminates the need to configure policy routes or static routes.

Configuring a QoS-Enforced Open Proxy Using RSVP Example

Figure 83 illustrates a proxy configuration that was created on a Cisco 2500 router with one Ethernet interface and two serial interfaces. Only the Ethernet interface is in use.

Figure 83 Configuring a QoS-Enforced Open Proxy Using RSVP



PX1 Configuration

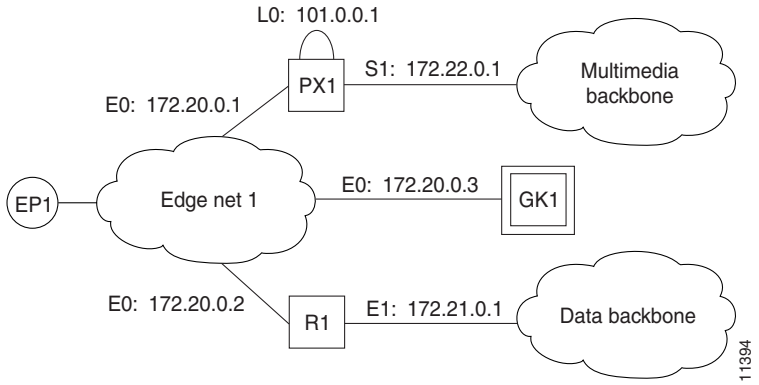
The following output is for the PX1 configuration:

```
!
version 11.3
no service password-encryption
service tcp-small-servers
!
hostname ExampleProxy
!
no ip domain-lookup
!
proxy h323
!
interface Ethernet0
 ip address 172.21.127.38 255.255.255.192
 no ip redirects
 ip rsvp bandwidth 7000 7000
 ip route-cache same-interface
 fair-queue 64 256 1000
 h323 interface
 h323 qos rsvp controlled-load
 h323 h323-id px1@zone1.com
 h323 gatekeeper ipaddr 172.21.127.39
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
router rip
 network 172.21.0.0
!
ip classless
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 password lab
 login
!
end
```

Configuring a Closed Co-Edge Proxy with ASR Without Subnetting Example

Figure 84 illustrates how to configure RIP on the edge networks and IGRP on the two backbone networks. A Cisco 2500 router is used for the proxy.

Figure 84 Configuring a Closed Co-Edge Proxy with ASR



PX1 Configuration

The following output is for the PX1 configuration:

```
!
version 11.3
no service password-encryption
service tcp-small-servers
!
hostname ExampleProxy
!
!
no ip domain-lookup
!
!
proxy h323
!
interface Loopback0
 ip address 10.0.0.1 255.0.0.0
 h323 interface
 h323 qos ip-precedence 4
 h323 h323-id px1@zone1.com
 h323 gatekeeper ipaddr 172.20.0.3
!
interface Ethernet0
 ip address 172.20.0.1 255.255.255.192
 no ip redirects
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 ip address 172.22.0.1 255.255.0.0
 ip access-group 101 in
 ip access-group 101 out
 h323 asr
!
router rip
```

```
 network 172.20.0.0
 network 10.0.0.0
!
router igrp 4000
 network 172.22.0.0
 network 101.0.0.0
!
ip classless
access-list 101 permit ip any host 10.0.0.1
access-list 101 permit ip host 10.0.0.1 any
access-list 101 permit igrp any any
!
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 password lab
 login
```

Defining Multiple Zones Example

The following example shows how to define multiple local zones for separating gateways:

```
zone local gk408or650 xyz.com
zone local gk415 xyz.com
zone prefix gk408or650 408.......
zone prefix gk408or650 650.......
zone prefix gk415 415.......
```

All the gateways used for area codes 408 or 650 can be configured so that they register with gk408or650, and all gateways used for area code 415 can be configured so that they register with gk415.

Defining One Zone for Multiple Gateways Example

The following example shows how to put all the gateways in the same zone and use the **gw-priority** keyword to determine which gateways will be used for calling different area codes:

```
zone local localgk xyz.com
zone prefix localgk 408.......
zone prefix localgk 415....... gw-priority 10 gw1 gw2
zone prefix localgk 650....... gw-priority 0 gw1
```

The commands shown accomplish the following tasks:

- Domain xyz.com is assigned to gatekeeper localgk.
- Prefix 408..... is assigned to gatekeeper localgk, and no gateway priorities are defined for it; therefore, all gateways that register to localgk can be used equally for calls to the 408 area code. No special gateway lists are built for the 408..... prefix; selection is made from the master list for the zone.
- The prefix 415..... is added to gatekeeper localgk, and priority 10 is assigned to gateways gw1 and gw2.
- Prefix 650..... is added to gatekeeper localgk, and priority 0 is assigned to gateway gw1.

A priority 0 is assigned to gateway gw1 to exclude it from the gateway pool for prefix 650..... When gateway gw2 registers with gatekeeper localgk, it is added to the gateway pool for each prefix as follows:

- For gateway pool for 415....., gateway gw2 is set to priority 10.
- For gateway pool for 650....., gateway gw2 is set to priority 5.

To change gateway gw2 from priority 10 for zone 415..... to the default priority 5, enter the following command:

```
no zone prefix localgk 415..... gw-pri 10 gw2
```

To change both gateways gw1 and gw2 from priority 10 for zone 415..... to the default priority 5, enter the following command:

```
no zone prefix localgk 415..... gw-pri 10 gw1 gw2
```

In the preceding example, the prefix 415..... remains assigned to gatekeeper localgk. All gateways that do not specify a priority level for this prefix are assigned a default priority of 5. To remove the prefix and all associated gateways and priorities from this gatekeeper, enter the following command:

```
no zone prefix localgk 415.....
```

Configuring a Proxy for Inbound Calls Example

In the following example, the local zone sj.xyz.com is configured to use a proxy for inbound calls from remote zones tokyo.xyz.com and milan.xyz.com to gateways in its local zone. The sj.xyz.com zone is also configured to use a proxy for outbound calls from gateways in its local zone to remote zones tokyo.xyz.com and milan.xyz.com.

```
gatekeeper
use-proxy sj.xyz.com remote-zone tokyo.xyz.com inbound-to gateway
use-proxy sj.xyz.com remote-zone tokyo.xyz.com outbound-from gateway
use-proxy sj.xyz.com remote-zone milan.xyz.com inbound-to gateway
use-proxy sj.xyz.com remote-zone milan.xyz.com outbound-from gateway
```

Because the default mode disables proxy communications for all gateway calls, only the gateway call scenarios listed can use the proxy.

Configuring a Proxy for Outbound Calls Example

In the following example, the local zone sj.xyz.com uses a proxy for only those calls that are outbound from H.323 terminals in its local zone to the specified remote zone germany.xyz.com:

```
gatekeeper
no use-proxy sj.xyz.com default outbound-from terminal
use-proxy sj.xyz.com remote-zone germany.xyz.com outbound-from
 terminal
```

Note that any calls inbound to H.323 terminals in the local zone sj.xyz.com from the remote zone germany.xyz.com use the proxy because the default applies.

Removing a Proxy Example

The following example shows how to remove one or more proxy statements for the remote zone germany.xyz.com from the proxy configuration list:

```
no use-proxy sj.xyz.com remote-zone germany.xyz.com
```

The command removes all special proxy configurations for the remote zone germany.xyz.com. After the command is entered like this, all calls between the local zone (sj.xyz.com) and germany.xyz.com are processed according to the defaults defined by any **use-proxy** commands that use the **default** option.

H.235 Security Example

The following example shows output from configuring secure registrations from the gatekeeper and identifying which RAS messages the gatekeeper will check to find authentication tokens:

```
dial-peer voice 10 voip
 destination-pattern 4088000
 session target ras
 dtmf-relay h245-alphanumeric
!
gateway
 security password 09404F0B level endpoint
```

The following example shows output from configuring which RAS messages will contain gateway generated tokens:

```
dialer-list 1 protocol ip permit

dialer-list 1 protocol ipx permit
radius-server host 10.0.0.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server deadtime 5
radius-server key lab
radius-server vsa send accounting
!
gatekeeper
 zone local GK1 test.com 10.0.0.3
 zone remote GK2 test2.com 10.0.2.2 1719
 accounting
 security token required-for registration
 no use-proxy GK1 remote-zone GK2 inbound-to terminal
 no use-proxy GK1 remote-zone GK2 inbound-to gateway
 no shutdown
```

GKTMP and RAS Messages Example

The following is an example of a gatekeeper that has interaction with external applications. The registration message from Server-123 establishes a connection with gatekeeper sj.xyz.com on port 20000. Server-123 sends a REGISTER RRQ message to gatekeeper sj.xyz.com to express interest in all RRQs from voice gateways that support a technology prefix of 1# or 2#.

```
REGISTER RRQ
Version-id:1
From:Server-123
To:sj.xyz.com
Priority:2
Notification-Only:
Content-Length:29

t=voice-gateway
p=1#
p=2#
```

Prohibiting Proxy Use for Inbound Calls Example

To prohibit proxy use for inbound calls to H.323 terminals in a local zone from a specified remote zone, enter a command similar to the following:

```
no use-proxy sj.xyz.com remote-zone germany.xyz.com inbound-to terminal
```

This command overrides the default and disables proxy use for inbound calls from remote zone germany.xyz.com to all H.323 terminals in the local zone sj.xyz.com.

Disconnecting a Single Call Associated with an H.323 Gateway Example

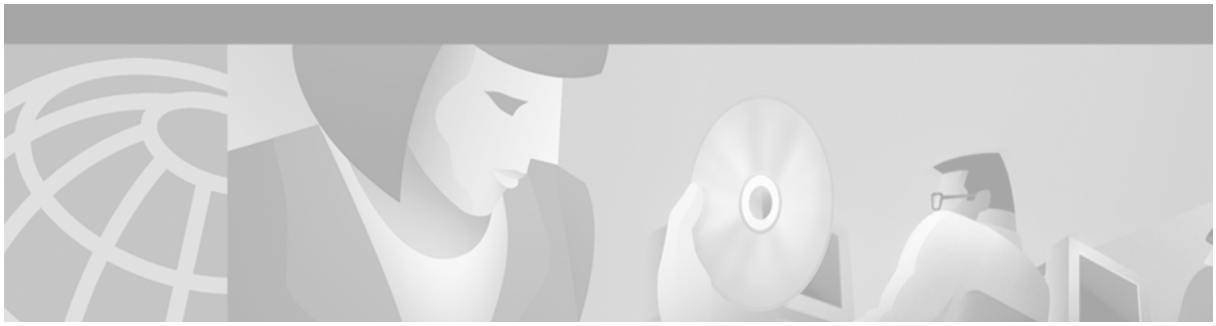
The following example forces an active call on the H.323 gateway to be disconnected. The local ID number of the active call is 12-3339.

```
Router> enable
Router# clear h323 gatekeeper call local-callID 12-3339
```

Disconnecting All Calls Associated with an H.323 Gateway Example

The following example forces all active calls on the H.323 gateway to be disconnected:

```
Router> enable
Router# clear h323 gatekeeper call all
```



Configuring Session Initiation Protocol for Voice over IP

This chapter introduces the Session Initiation Protocol (SIP). SIP is an alternative protocol developed by the Internet Engineering Task Force (IETF) for multimedia conferencing over IP. SIP features are compliant with IETF RFC 2543, SIP: Session Initiation Protocol, published in March 1999.

The Cisco SIP functionality enables Cisco access platforms to signal the setup of voice and multimedia calls over IP networks. The SIP feature also provides nonproprietary advantages in the following areas:

- Protocol extensibility
- System scalability
- Personal mobility services
- Interoperability with different vendors

This chapter contains the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" in the "Using Cisco IOS Software" chapter.

SIP Overview

SIP is an ASCII-based, application-layer control protocol that can be used to establish, maintain, and terminate calls between two or more endpoints.

Like other Voice over IP protocols, SIP is designed to address the functions of signaling and session management within a packet telephony network. Signaling allows call information to be carried across network boundaries. Session management provides the ability to control the attributes of an end-to-end call.

SIP provides the following capabilities:

- Determines the location of the target endpoint—SIP supports address resolution, name mapping, and call redirection.
- Determines the media capabilities of the target endpoint—Through Session Description Protocol (SDP), SIP determines the lowest level of common services between the endpoints. Conferences are established using only the media capabilities that can be supported by all endpoints.
- Determines the availability of the target endpoint—If a call cannot be completed because the target endpoint is unavailable, SIP determines whether the called party is connected to a call already or did not answer in the allotted number of rings. SIP then returns a message indicating why the target endpoint was unavailable.
- Establishes a session between the originating and target endpoints—If the call can be completed, SIP establishes a session between the endpoints. SIP also supports midcall changes, such as the addition of another endpoint to the conference or the changing of a media characteristic or codec.
- Handles the transfer and termination of calls—SIP supports the transfer of calls from one endpoint to another. During a call transfer, SIP simply establishes a session between the transferee and a new endpoint (specified by the transferring party) and terminates the session between the transferee and the transferring party. At the end of a call, SIP terminates the sessions among all parties.

Note The term conference means an established session (or call) between two or more endpoints. Conferences consist of two or more users and can be established using multicast or multiple unicast sessions.

Components of SIP

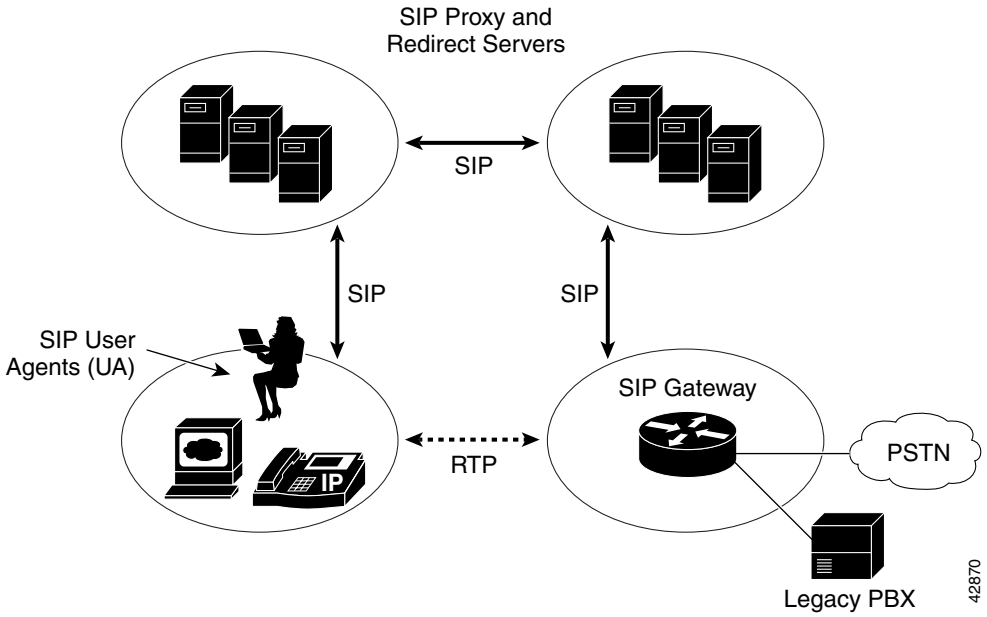
SIP is a peer-to-peer protocol. The peers in a session are called user agents (UAs). A user agent can function in one of the following roles:

- User agent client (UAC)—A client application that initiates the SIP request.
- User agent server (UAS)—A server application that contacts the user when a SIP request is received and that returns a response on behalf of the user.

Typically, a SIP endpoint is capable of functioning as both a UAC and a UAS, but functions only as one or the other per transaction. Whether the endpoint functions as a UAC or a UAS depends on the UA that initiated the request.

From an architectural standpoint, the physical components of a SIP network can be grouped into two categories: clients and servers. Figure 85 illustrates the architecture of a SIP network.

Figure 85 SIP Architecture



Note The SIP servers can interact with other application services, such as Lightweight Directory Access Protocol (LDAP) servers, location servers, a database application, or an extensible markup language (XML) application. These application services provide back-end services such as directory, authentication, and billing services.

SIP Clients

SIP clients include the following:

- Phones—Can act as either a UAS or UAC. SoftPhones (PCs that have phone capabilities installed) and Cisco SIP IP phones can initiate SIP requests and respond to requests.
- Gateways—Provide call control. Gateways provide many services, the most common being a translation function between SIP conferencing endpoints and other terminal types. This function includes translation between transmission formats and between communications procedures. In addition, the gateway translates between audio and video codecs and performs call setup and clearing on both the LAN side and the switched-circuit network side.

SIP Servers

SIP servers include the following:

- Proxy server—Receives SIP messages and forwards them to the next SIP server in the network. The proxy server is an intermediate device that receives SIP requests from a client and then forwards the requests on behalf of the client. Proxy servers can provide functions such as authentication, authorization, network access control, routing, reliable request retransmission, and security.
- Redirect server—Provides the client with information about the next hop or hops that a message should take. The client then contacts the next hop server or UAS directly.
- Registrar server—Processes requests from UACs for registration of their current location. Registrar servers are often located near a redirect or proxy server.

How SIP Works

SIP is a simple, ASCII-based protocol that uses requests and responses to establish communication among the various components in a network and ultimately to establish a conference between two or more endpoints.

Users in a SIP network are identified by unique SIP addresses. A SIP address is similar to an e-mail address and is in the format of sip:*userID@gateway*.com. The user ID can be either a username or an E.164 address.

Users register with a registrar server using their assigned SIP addresses. The registrar server then provides the registration information to the location server upon request.

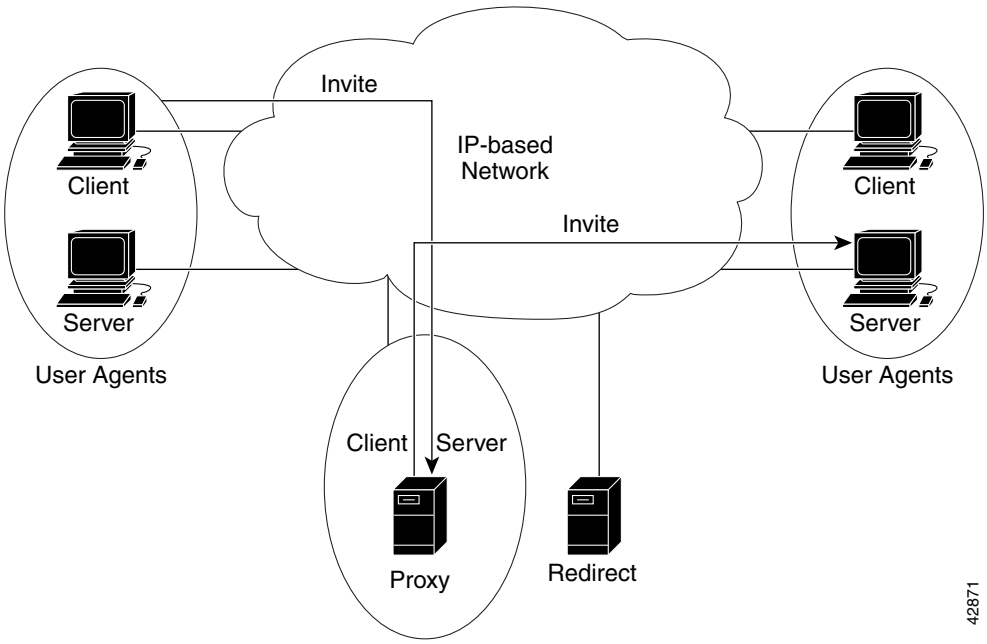
When a user initiates a call, a SIP request is sent to a SIP server (either a proxy or a redirect server). The request includes the address of the caller (in the “from” header field) and the address of the intended callee (in the “to” header field). The following sections provide simple examples of successful point-to-point calls established using a proxy and a redirect server.

Over time, a SIP end user might move between end systems. The location of the end user can be dynamically registered with the SIP server. The location server can use one or more protocols (including finger, rwhois, and LDAP) to locate the end user. Because the end user can be logged in at more than one station and because the location server can sometimes have inaccurate information, the SIP server might return more than one address for the end user. If the request is coming through a SIP proxy server, the proxy server will try each of the returned addresses until it locates the end user. If the request is coming through a SIP redirect server, the redirect server forwards all the addresses to the caller in the “contact” header field of the invitation response.

Using a Proxy Server

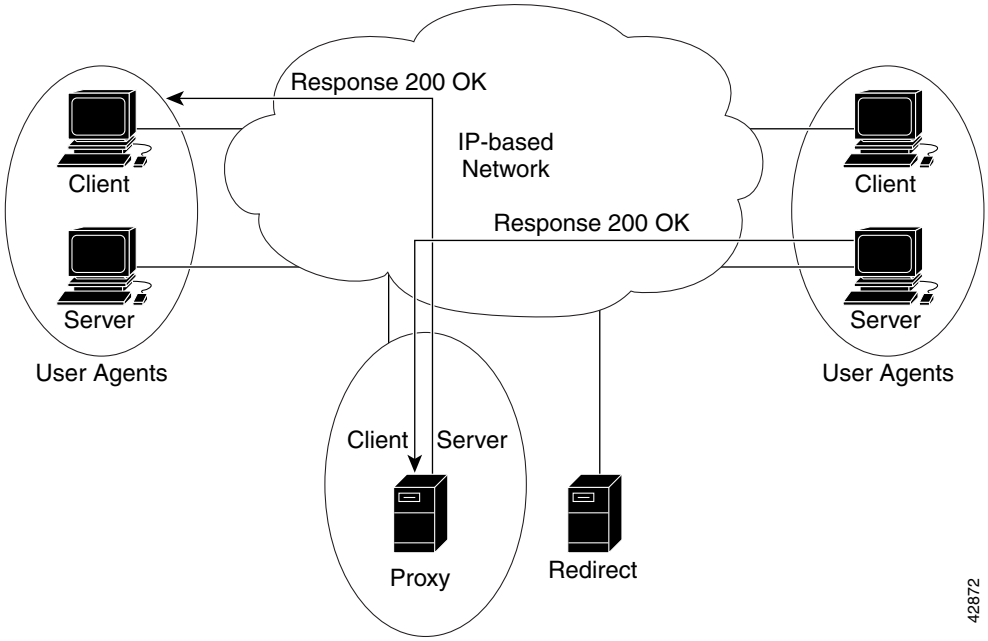
If a proxy server is used, the caller UA sends an INVITE request to the proxy server. The proxy server determines the path and then forwards the request to the callee, as shown in Figure 86.

Figure 86 SIP Request Through a Proxy Server



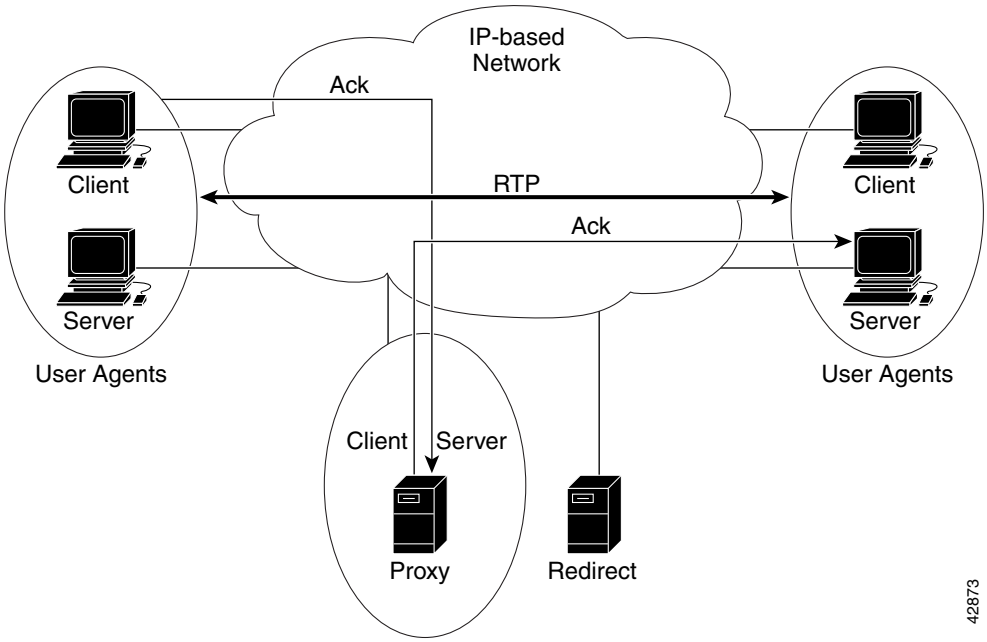
The callee responds to the proxy server, which in turn forwards the response to the caller, as shown in Figure 87.

Figure 87 SIP Response Through a Proxy Server



The proxy server forwards the acknowledgments of both parties. A session is then established between the caller and callee. Real-Time Transfer Protocol (RTP) is used for the communication between the caller and the callee, as shown in Figure 88.

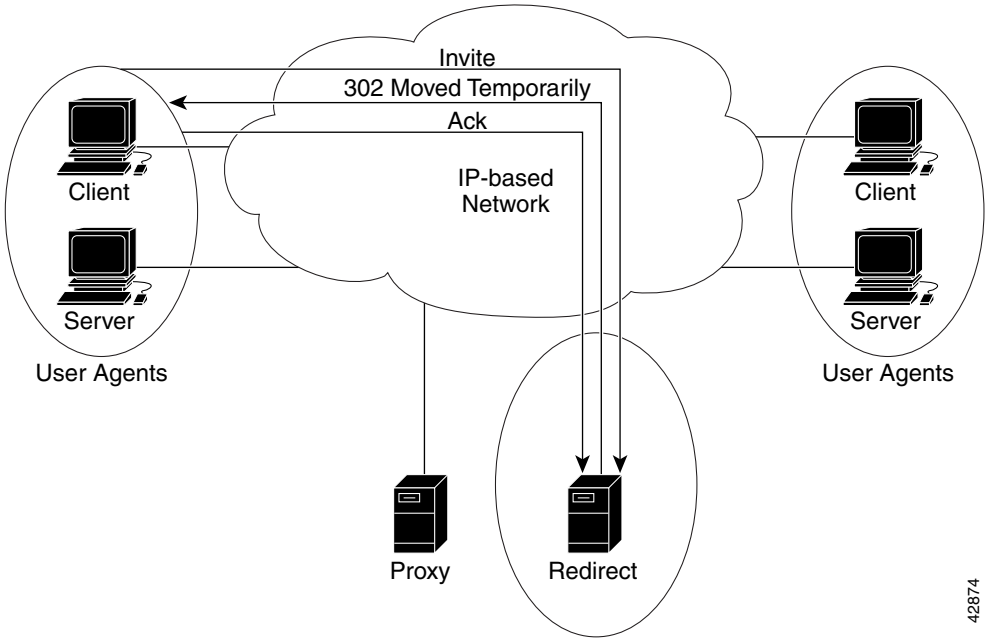
Figure 88 SIP Session Through a Proxy Server



Using a Redirect Server

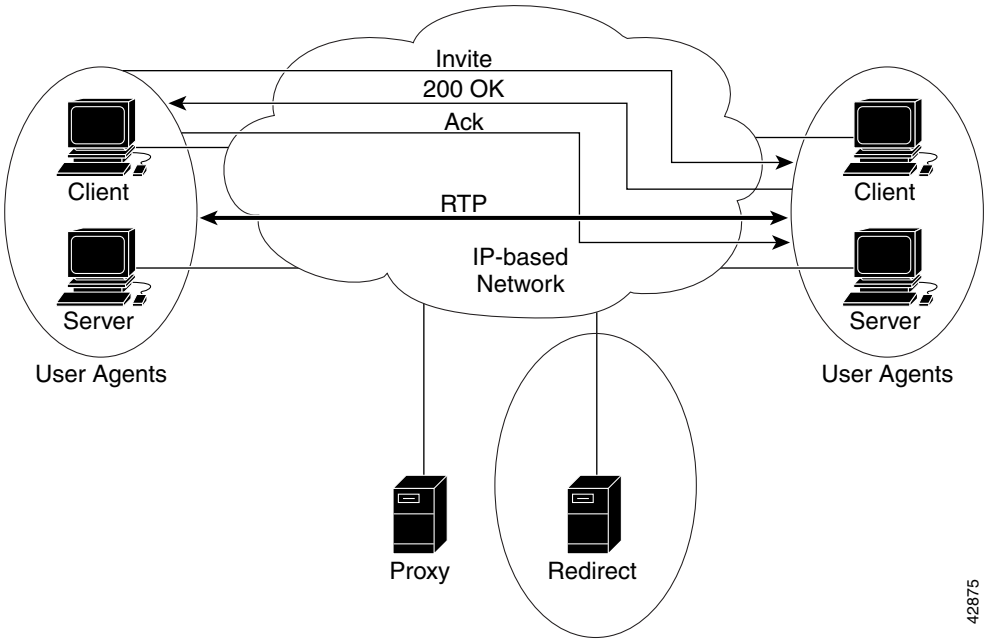
If a redirect server is used, the caller UA sends an INVITE request to the redirect server. The redirect server contacts the location server to determine the path to the callee, and the redirect server sends that information back to the caller. The caller then acknowledges receipt of the information, as shown in Figure 89.

Figure 89 SIP Request Through a Redirect Server



The caller then sends a request to the device indicated in the redirection information (which could be the callee or another server that will forward the request). Once the request reaches the callee, it sends back a response, and the caller acknowledges the response. RTP is used for the communication between the caller and the callee, as shown in Figure 90.

Figure 90 SIP Session Through a Redirect Server



SIP Enhancements

SIP provides the following feature enhancements:

- Ability to specify the maximum number of SIP redirects.
- Ability to specify SIP or H.323 on a dial-peer basis.
- Configurable SIP message timers and retries.
- Interoperability with unified call services (UCS).
- Support for a variety of signaling protocols, including ISDN, PRI, and channel associated signaling (CAS).
- Support for a variety of interfaces, including
  - Analog interfaces: Foreign Exchange Station (FXS)/Foreign Exchange Office (FXO)/recEive and transMit (E&M) analog interfaces.
  - Digital interfaces: T1 CAS, T1 PRI, E1 CAS, E1 PRI, and E1 R2
- Support for SIP redirection messages and interaction with SIP proxies. The gateway can redirect an unanswered call to another SIP gateway or SIP-enabled IP phone. In addition, the gateway supports proxy-routed calls.
- Interoperability with DNS servers, including support for DNS SRV and "A" records to look up SIP URLs according to RFC2052 formatting.
- Support for SIP over TCP and User Datagram Protocol (UDP).
- Support RTP/RTCP for media transport in VoIP networks.
- Support for the following codecs:
  - G711ulaw—0
  - G711alaw—8
  - G723r63—4
  - G726r32—2
  - G728—15
  - G729r8—18
- Support for record-route headers.
- Support for IP quality of service (QoS) and IP precedence.
- Support for IP Security (IPSec) for SIP signaling messages.
- Authentication, authorization, and accounting (AAA) support. For accounting, the gateway device generates call data record (CDR) accounting records for export. For authentication, the SIP gateway sends validation requests to the AAA server. For authorization, the existing access lists are used.
- Support for call hold and call transfer features. The call hold sends a midcall INVITE message, which requests that the remote endpoint stop sending media streams. The call transfer is done without consultation (blind transfer). The transfer can be initiated by a remote SIP endpoint.
- Support for configurable expiration time for SIP INVITEs and maximum number of proxies or redirect servers that can forward a SIP request.
- Ability to hide the identity of the calling party by setting the ISDN presentation indicator.

SIP Restrictions and Considerations

Before configuring your router (Cisco 2600, Cisco 3600, or Cisco AS5300) with the SIP feature, you should note the following restrictions and considerations:

- The SIP gateway does not support codecs other than those listed in the section, "SIP Enhancements."
- SIP requires that all times be sent in Greenwich Mean Time (GMT). The INVITE is sent in GMT. However, the default for routers is to use Coordinated Universal Time (UTC). To configure the router to use GMT, issue the **clock timezone** command in global configuration mode and specify GMT.
- With call transfer, the Requested-By header identifies the party initiating the transfer. The Requested-By header is included in the INVITE request that is sent to the transferred-to party only if a Requested-By header was also included in the Bye request.
- With call transfer, the Also header identifies the transferred-to party. To invoke a transfer, the user portion of the Also header must be defined explicitly or with wildcards as a destination pattern on a VoIP dial peer. The transferred call is routed using the session target parameter on the dial peer instead of the host portion of the Also header. Therefore, the Also header can contain *user@host*, but the *host* portion is ignored for call routing purposes.
- The grammar for the Also and Requested-By headers is not fully supported. Only the name-addr is supported. This implies that the crypto-param, which might be present in the Bye request, will not be populated in the ensuing Invite to the transferred-to party.
- Cisco SIP gateways do not support the "user=np-queried" parameter in a Request URI.
- If a Cisco SIP gateway receives an ISDN Progress message, it generates a 183 Session progress message. If the gateway receives an ISDN ALERT, it generates a 180 Ringing message.
- SIP supports plain old telephone service (POTS)-to-POTS hairpinning (which means that the call comes in one voice port and is routed out another voice port). It also supports POTS-to-IP call legs and IP-to-POTS call legs. However, it does not support IP-to-IP hairpinning. This means that the SIP gateway cannot take an inbound SIP call and reroute it back to another SIP device using the VoIP dial peers.
- The SIP gateway requires each INVITE to include a Session Description Protocol (SDP) header.
- The contents of the SDP header cannot change between the 180 Ringing message and the 200 OK message.
- VoIP dial peers allow a user to configure the **bytes** parameter associated with a codec. However, Cisco SIP gateways do not present or respond to this parameter. The **a=ptime** parameter is not sent or recognized in the SDP body of a SIP message.

SIP Prerequisite Tasks

Before you configure your router with the SIP feature, you must perform the following tasks:

- Configure your gateway to support voice functionality for SIP or H.323.
- Establish a working IP network.

  For more information about configuring IP, refer to the *Cisco IOS IP Configuration Guide*.
- Configure VoIP.
- Ensure that your Cisco 2600 or Cisco 3600 series router has16 MB Flash and 64 MB DRAM memory, minimum. A Cisco AS5300 must have 16 MB Flash and 64 MB DRAM memory, minimum.

SIP Configuration Tasks List

To configure SIP functions on the Cisco AS5300, Cisco 2600, or the Cisco 3600 series router, perform the following tasks:

- Configuring SIP Support for VoIP Dial Peers, page 412
- Changing the Configuration of the SIP User Agent, page 413 (Optional)
- Configuring SIP Call Transfer, page 414 (Optional)
- Configuring Gateway Accounting, page 415 (Optional)

For more information on SIP configuration, including call flows, refer to the document *Session Initiation Protocol Gateway Call Flows, Version 2* in Cisco IOS Release 12.1(3)T found on Cisco.com.

Configuring SIP Support for VoIP Dial Peers

To configure SIP support for a VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters dial-peer configuration mode to configure a VoIP dial peer.
Step 2	`Router(config-dial-peer)# session transport {udp | tcp}`	Enters the session transport type for the SIP user agent. The default is **udp**. The transport protocol (**udp** or **tcp**) specified with the **session transport** command must be identical to the protocol specified with the **transport** command.
Step 3	`Router(config-dial-peer)# session protocol {cisco | sipv2}`	Enters the session protocol type. The keywords are as follows: • **cisco**—Configures the dial peer to use proprietary CiscoVoIP session protocol. • **sipv2**—Configures the dial peer to use IETF SIP. SIP users should use this option.

	Command	Purpose
Step 4	`Router(config-sip-ua)# sip-server {dns:[hostname] | ipv4:ip_addr:[port-num]}`	Enters the host name or IP address of the SIP server interface. If you use this command, you can then specify **session target sip-server** for each dial peer instead of repeatedly entering the SIP server interface address for each dial peer. The keywords and arguments are as follows: • **dns:***hostname*—Sets the global SIP server interface to a domain name server (DNS) host name. A valid DNS host name takes the following format: *name.gateway*.*xyz*. • **ipv4:***ip_addr:*—Sets the IP address. • *portnum* (Optional)—Sets the UDP port number for the SIP server.
Step 5	`Router(config-dial-peer)# session target {sip-server | dns:[$s$. | $d$. | $e$. | $u$. [hostname]| ipv4:ip_addr:[port-num]}`	Specifies a network-specific address for a dial peer. The keywords and arguments are as follows: • **sip-server**— Sets the session target to the global SIP server. Used when the sip-server command has already specified the host name or IP address of the SIP server interface. • **dns:***hostname*—Sets the global SIP server interface to a domain name server (DNS) host name. A valid DNS host name takes the following format: *name.gateway*.*xyz*. • **ipv4:***ip_addr:*—Sets the IP address. • *portnum*—(Optional) Sets the UDP port number for the SIP server. **Note** Wildcards can be used when defining the session target for VoIP peers.

Changing the Configuration of the SIP User Agent

It is not necessary to configure a SIP user agent (UA) in order to place a call. A SIP UA is configured to listen by default. However, if you want to adjust any of the settings, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# sip-ua`	Enters the SIP user agent (sip-ua) configuration mode to configure SIP-UA related commands.
Step 2	`Router(config-sip-ua)# transport {udp | tcp}`	Configures the SIP user agent (sip-ua) for SIP signaling messages. The default is **udp**. The transport protocol (**udp** or **tcp**) specified with the **session transport** command must be identical to the protocol specified with the **transport** command.

	Command	Purpose
Step 3	`Router(config-sip-ua)# timers {trying number | connect number | disconnect number | expires number}`	(Optional) Configures the SIP signaling timers. The keywords are as follows: • **trying**—Sets the time to wait for a 100 response to an INVITE request. The default is 500. • **connect**—Sets the time to wait for a 200 response to an ACK request. The default is 500. • **disconnect**—Sets the time to wait for a 200 response to a BYE request. The default is 500. • **expires**—Limits the time duration (in milliseconds) for which an INVITE is valid. The default is 180000.
Step 4	`Router(config-sip-ua)# retry {invite number | response number | bye number | cancel number}`	(Optional) Configures the SIP signaling timers for retry attempts. The keywords are as follows: • **invite**—Number of INVITE retries. The default is 6. • **response**—Number of RESPONSE retries. The default is 6. • **bye**—Number of BYE retries. The default is 10. • **cancel**—Number of Cancel retries. The default is 10.
Step 5	`Router(config-sip-ua)# max-forwards number`	(Optional) Limits the number of proxy or redirect servers that can forward a request. The default is 6.
Step 6	`Router(config-sip-ua)# max-redirects number`	(Optional) Sets the maximum number of redirect servers. The default is 1.
Step 7	`Router(config-sip-ua)# default {max-forwards | retry {invite | response | bye | cancel} | sip-server | timers {trying | connect | disconnect | expires} | transport}`	(Optional) Resets the value of a SIP user agent command to its default.

Configuring SIP Call Transfer

To configure SIP call transfer for a POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number pots`	Enters dial-peer configuration mode to configure a POTS dial peer.
Step 2	`Router(config-dial-peer)# application session`	Specifies that the standard session application will be invoked for this dial peer.
Step 3	`Router(config-dial-peer)# destination-pattern pattern`	Specifies the telephone number associated with the dial peer.
Step 4	`Router(config-dial-peer)# port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	(Cisco 2600 and Cisco 3600 series routers) Specifies the local voice port through which incoming VoIP calls will be received.
Step 5	`Router(config-dial-peer)# port {controller number:D}`	(Cisco AS5300 universal access server) Specifies the local voice port through which incoming VoIP calls will be received.

To configure SIP call transfer for a VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters the dial-peer mode to configure a VoIP dial peer.
Step 2	`Router(config-dial-peer)# application session`	Specifies that the standard session application will be invoked for this dial peer.
Step 3	`Router(config-dial-peer)# destination-pattern` *pattern*	Specifies the telephone number associated with the dial peer.
Step 4	`Router(config-dial-peer)# session target ipv4:`*x.x.x.x*	Specifies the IP address of the destination gateway for outbound dial peers.

Note For information about the commands used to configure translation rules, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

Configuring Gateway Accounting

There are three keywords that configure gateway accounting for SIP:

- The **voip** keyword sends the call data record (CDR) to the RADIUS server. Use this keyword with the SIP feature.
- The **H323** keyword sends the call data record (CDR) to the RADIUS server.
- The **syslog** keyword uses the system logging facility to record the CDRs.

To enable gateway-specific accounting for SIP, use the following command in global configuration mode:

Command	Purpose
`Router(config)# gw-accounting {voip | syslog | h323 [syslog]}`	(Optional) Enables gateway-specific accounting in global configuration mode.

For general accounting information, refer to the *Cisco IOS Security Configuration Guide*.

Verifying SIP Configuration

Enter the **show running-config** command to verify your configuration, or use the **show sip-ua** command to verify the SIP configurations.

The following example shows sample output for the **show sip-ua statistics** command:

```
Router# show sip-ua statistics

SIP Response Statistics (Inbound/Outbound)
    Informational:
      Trying 0/0, Ringing 0/0,
      Forwarded 0/0, Queued 0/0,
      SessionProgress 0/0
    Success:
       OkInvite 0/0, OkBye 0/0,
       OkCancel 0/0, OkOptions 0/0
    Redirection (Inbound only):
      MultipleChoice 0, MovedPermanently 0,
      MovedTemporarily 0, SeeOther 0,
      UseProxy 0, AlternateService 0
    Client Error:
      BadRequest 0/0, Unauthorized 0/0,
      PaymentRequired 0/0, Forbidden 0/0,
      NotFound 0/0, MethodNotAllowed 0/0,
      NotAcceptable 0/0, ProxyAuthReqd 0/0,
      ReqTimeout 0/0, Conflict 0/0, Gone 0/0,
      LengthRequired 0/0, ReqEntityTooLarge 0/0,
      ReqURITooLarge 0/0, UnsupportedMediaType 0/0,
      BadExtension 0/0, TempNotAvailable 0/0,
      CallLegNonExistent 0/0, LoopDetected 0/0,
      TooManyHops 0/0, AddrIncomplete 0/0,
      Ambiguous 0/0, BusyHere 0/0
    Server Error:
      InternalError 0/0, NotImplemented 0/0,
      BadGateway 0/0, ServiceUnavail 0/0,
      GatewayTimeout 0/0, BadSipVer 0/0
    Global Failure:
      BusyEverywhere 0/0, Decline 0/0,
      NoExistAnywhere 0/0, NotAcceptable 0/0

SIP Total Traffic Statistics (Inbound/Outbound)
    Invite 0/0, Ack 0/0, Bye 0/0,
    Cancel 0/0, Options 0/0

Retry Statistics
    Invite 0, Bye 0, Cancel 0, Response 0
```

The following example shows sample output for the **show sip-ua status** command:

```
Router# show sip-ua status

SIP User Agent Status
SIP User Agent for UDP : ENABLED
SIP User Agent for TCP : ENABLED
SIP max-forwards :6
```

The following example shows sample output for the **show sip-ua timers** command:

```
Router# show sip-ua timers

SIP UA Timer Values (millisecs)
trying 500, expires 180000, connect 500, disconnect 500
```

SIP Configuration Examples

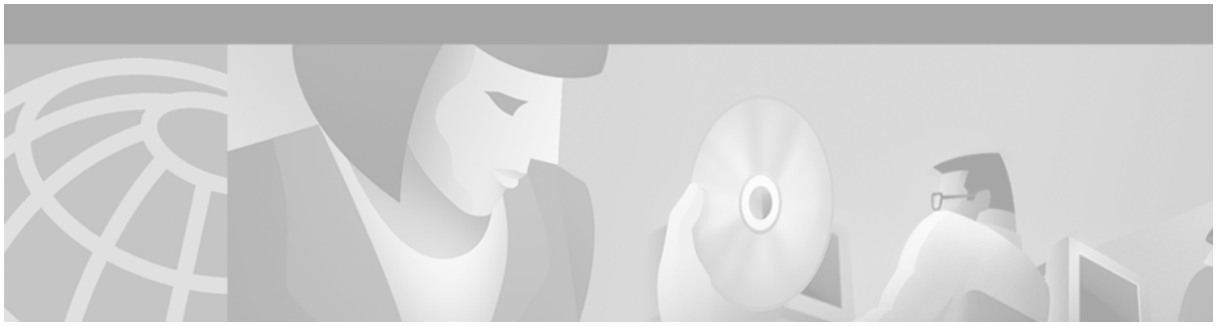
The following shows a basic SIP configuration. This output was created by using the **show running-config** command.

```
Router1# show running-config

Building configuration...

Current configuration:
!
version 12.2
service timestamps debug datetime
service timestamps log uptime
no service password-encryption
!
hostname router1
!
!
!
clock timezone GMT 5
voice-card 1
!
ip subnet-zero
ip tcp path-mtu-discovery
ip name-server 172.18.192.48
!
isdn voice-call-failure 0
!
!
controller T1 1/0
 framing esf
 clock source line primary
 linecode b8zs
!
controller T1 1/1
!
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice class codec 1
 codec preference 1 g711alaw
 codec preference 2 g723r63
 codec preference 3 g723r53
!
!
dial-peer voice 100 pots
 destination-pattern 3660110
 port 2/0/0
!
dial-peer voice 200 pots
 application session
 destination-pattern 3660120
 port 2/0/1
!
dial-peer voice 101 voip
 destination-pattern 3660210
 session protocol sipv2
 session target ipv4:172.16.244.73
 codec g711ulaw
!
```

```
dial-peer voice 201 voip
 application session
 destination-pattern 3660220
 session protocol sipv2
 session target dns:3660-2.sip.com
 codec g711ulaw
!
dial-peer voice 999 voip
 destination-pattern 5551111
 session protocol sipv2
 session target ipv4:172.20.53.89
 session transport tcp
!
dial-peer voice 300 pots
 destination-pattern 2101100
!
dial-peer voice 350 voip
 destination-pattern 3100607
 session protocol sipv2
 session target ipv4:172.18.192.197
 codec g711ulaw
!
dial-peer voice 301 voip
 application session
 destination-pattern 1234
 session protocol sipv2
 session target ipv4:172.18.192.193
 codec g711ulaw
!
dial-peer voice 333 voip
 application session
 destination-pattern 1235
 session protocol sipv2
 session target ipv4:172.18.192.199
 codec g711ulaw
!
dial-peer voice 888 voip
 destination-pattern 888
 session protocol sipv2
 session target ipv4:172.20.53.89
 session transport tcp
 codec g711ulaw
!
dial-peer voice 260011 voip
 destination-pattern 260011
 session protocol sipv2
 session target ipv4:172.18.192.164
 codec g711ulaw
!
dial-peer voice 444 voip
 destination-pattern 2339000
 session protocol sipv2
 session target ipv4:172.18.192.205
 codec g711ulaw
!
dial-peer voice 111 voip
 destination-pattern 111
 session protocol sipv2
 session target sip-server
 codec g711ulaw
!
dial-peer voice 7777777 voip
 destination-pattern 19197777777
 session protocol sipv2
```

```
 session target ipv4:172.18.192.38
 codec g711ulaw
!
!
sip-ua
retry invite 2
retry response 2
retry bye 2
retry cancel 2
no inband-alerting
sip-server dns:server
!
!
interface FastEthernet0/0
 ip address 172.18.192.194 255.255.255.0
 load-interval 30
 speed auto
 half-duplex
!
interface FastEthernet0/1
 ip address 172.16.245.230 255.255.255.224
 load-interval 30
 speed auto
 half-duplex
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.192.1
ip route 172.16.0.0 255.255.0.0 172.16.245.225
no ip http server
!
access-list 101 permit ip host 10.0.2.30 host 10.0.2.31
access-list 101 deny   udp any eq rip any
access-list 101 deny   udp any any eq rip
access-list 101 deny   udp any eq isakmp any
access-list 101 deny   udp any any eq isakmp
access-list 101 permit ip any any
snmp-server engineID local 000000090200003094202740
snmp-server community public RW
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password xxx
 login
!
end
```

Voice over Layer 2 Protocols



Configuring Voice over Frame Relay

This chapter describes the configuration of Voice over Frame Relay (VoFR) and contains the following sections:

For a description of the VoFR configuration commands using the FRF.11 implementation agreement, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. For additional information about the FRF.12 implementation agreement and wide-area networks (WANs), refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and *Cisco IOS Wide-Area Networking Command Reference*. For information about voice port configurations, refer to the "Configuring Voice Ports" chapter.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" in the "Using Cisco IOS Software" chapter.

VoFR Overview

VoFR enables a router to carry voice traffic (for example, telephone calls and faxes) over a Frame Relay network, using the FRF.11 protocol. This specification defines multiplexed data, voice, fax, dual tone multi frequency (DTMF) digit-relay, and channel-associated signaling (CAS)/robbed-bit signaling frame formats. The Frame Relay backbone must be configured to include the map class and Local Management Interface (LMI).

The Cisco VoFR implementation enables dynamic- and tandem-switched calls and Cisco trunk calls. Dynamic-switched calls have dial-plan information included that processes and routes calls based on the telephone numbers. The dial-plan information is contained within dial-peer entries. For more information, see "Switched Calls" section on page 425.

Tandem-switched calls are switched from incoming VoFR to an outgoing VoFR enabled data-link connection identifier (DLCI) and tandem nodes enable the process. The nodes also switch Cisco trunk calls.

Permanent calls are processed over Cisco private-line trunks and static FRF.11 trunks that specify the frame format and coder types for voice traffic over a Frame Relay network. For more information, see "Permanent Calls" section on page 426.

VoFR connections depend on the hardware platform and type of call. The types of calls are:

- Switched (user dialed or auto-ringdown and tandem)
- Permanent (Cisco trunk or static FRF.11 trunk)

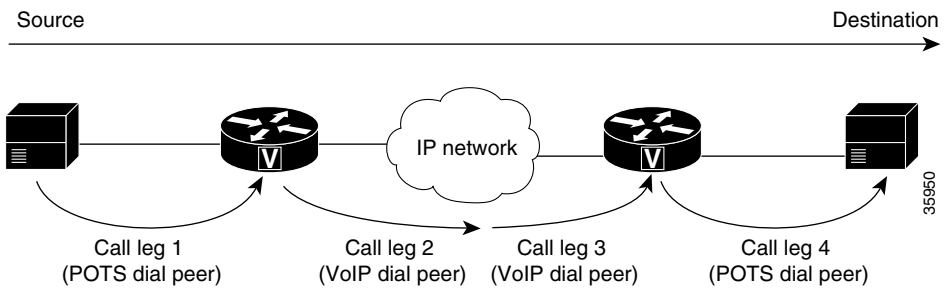
Note Calls to Cisco MC3810 multiservice concentrators running Cisco IOS releases before 12.0(7)XK and 12.1(2)T require specific procedures for VoFR configuration and are described in separate sections.

VoFR Dial Peers

Dial peers are addressable call endpoints that identify the origin and destination of a call. Dial peers define the characteristics applied to each call leg in the call connection. A call leg is a logical connection between two routers or between a router and a telephony device.

A traditional voice call over the Public Switched Telephone Network (PSTN) uses a dedicated 64K circuit end-to-end. In contrast, a voice call over the packet network is made up of call legs. A voice call has four call legs, two from the perspective of the originating router and two from the perspective of the destination router, as shown in Figure 91.

Figure 91 Dial Peer Call Legs



A dial peer is associated with each call leg. Attributes that are defined in a dial peer and applied to the call leg include codec, Quality of Service (QoS), voice activity detection (VAD), and fax rate. To complete a voice call, you must configure a dial peer for each of the four call legs in the call connection.

Two kinds of dial peers are possible in VoFR configurations:

- POTS—Dial peer describing the characteristics of a traditional telephony network connection. POTS dial peers map a dialed string to a specific voice port on the local router, normally the voice port connecting the router to the local PSTN, PBX, or telephone.
- VoFR—Dial peer that is connected between a Frame Relay WAN backbone and a specific voice-network device. VoFR dial peers map a dialed string to the destination router.

VoFR peers point to specific voice-network devices by associating destination telephone numbers with a specific Frame Relay DLCI so that outgoing calls can be placed. Both POTS and VoFR dial peers are needed to establish VoFR connections if the sending and receiving of calls are required.

Understanding the the relationship between the destination pattern and the session target is critical to understanding VoFR dial peers. The destination pattern is the telephone number of the voice device attached to the voice port. The session target defines the route to a serial port on the peer router at the other end of the Frame Relay connection.

Note For tandem voice nodes, POTS dial peers are not configured.

For additional information on POTS dial peers, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

Switched Calls

The Cisco-switched VoFR protocol handles call setup and parameter negotiation for both endpoints and intermediate nodes within the multihop call path. The call setup mechanism originally implemented in the Cisco MC3810 multiservice concentrator can be used for permanent-switched (Cisco trunk) or dynamic-switched calls. The Cisco VoFR protocol includes forwarding of the called telephone number and supports tandem switching of the call over multiple Frame Relay permanent virtual connection (PVC) hops.

Cisco addresses the lack of end-to-end call parameter negotiation and call setup syntax in FRF.11 by implementing a proprietary Q.931-like session protocol running on a user-configurable channel ID (CID) of an FRF.11-format multiplexed DLCI.

Tandem Switching

Dynamic switching of voice calls between VoFR or VoATM PVCs and subchannels is also called tandem switching (often encountered in multihop VoFR call connection paths). Tandem switching uses nodes that are intermediate router nodes within the Frame Relay call path.

Each node switches the frames from one PVC subchannel to another (from one VoFR dial peer to another VoFR dial peer) as the frames traverse the network. Use of tandem router nodes avoids the need to have complete dial-plan information present on every router.

Dynamic-Switched Calls

Dynamic-switched calls are regular telephone calls in which the switching is performed by the Cisco router. The destination endpoint of the call is selected by the router based on the dialed telephone number and the dial peer configuration entries. This implementation is different from permanent calls (Cisco trunk calls) in which the call endpoints are permanently fixed at configuration time. The dial peer uses the Cisco proprietary session protocol.

Cisco Trunk Calls

A Cisco trunk call is a dynamic-switched call of indefinite duration that uses a fixed-destination telephone number and includes optional transparent end-to-end signaling. The telephone number of the destination endpoint is permanently configured into the router so that it always selects a fixed destination. Once established, at boot-up or when configured, the call stays up until one of the voice ports or network ports is shut down or until a network disruption occurs. The dial peer is configured to invoke the Cisco proprietary session protocol.

Permanent Calls

Permanent calls are transmitted and received on FRF.11 and Cisco trunks. FRF.11 trunk interoperability for standards-based vendors enables specification of the frame format and coder types to be used when sending voice traffic through a Frame Relay network. However, FRF.11 does not have specifications for end-to-end negotiation, call setup process, or any other form of communication between the Frame Relay nodes.

As a result, static FRF.11 trunks are set up by manually configuring each router within the voice trunk path with compatible parameters: a voice port and a specific subchannel on a DLCI are explicitly bound on each end router. Signaling information is packed and sent transparently end-to-end.

The two ends of an FRF.11 call must use the same compatible speech compression codecs. If not, the call exists and voice packets are sent and received, but no usable voice path is created.

When configured, a static FRF.11 trunk remains up until the voice or serial port is shut down or until a network disruption occurs. The FRF.11 specification does not include any standardized methods for performing Operation, Administration, and Maintenance (OAM) functions. There is no standard protocol for detecting faults and providing rerouting of connection paths.

FRF.11 enables up to 255 subchannels to be multiplexed onto a single Frame Relay DLCI. The current implementation supports the multiplexing of a single data channel with many voice channels. However, subchannels from zero to three are reserved and cannot be configured for voice or data.

Frame Relay Fragmentation

Cisco has developed three methods of performing Frame Relay fragmentation that are described in the following sections:

- End-to-End FRF.12 Fragmentation, page 427
- Frame Relay Fragmentation Using FRF.11 Annex C, page 428
- Cisco Proprietary Voice Encapsulation, page 428

FRF.11 can only be used when an end-to-end PVC is available between the voice ports at each end of the connection. At intermediate Frame Relay nodes, the entire PVC must be routed. Because the entire PVC is routed, no prioritization of voice packets is possible at the intermediate Frame Relay. Connection ID-based routing (individual channel-ID switching) is not supported.

FRF.11 specifies that a device can pack multiple FRF.11 subframes within a single Frame Relay frame; however, the Cisco implementation of VoFR currently does not support multiple subframes within a frame. VoFR frames are never fragmented, regardless of size. If fragments arrive out of sequence, packets are dropped. Fragmentation is performed after frames are removed from the weighted fair queuing (WFQ). WFQ at the PVC level is the only queueing strategy that can be used.

Frame Relay Traffic Shaping (FRTS) must be configured to enable Frame Relay fragmentation.

Frame Relay fragmentation can be configured in conjunction with VoFR or independently of it. For additional information regarding FRF.12 fragmentation and the implementation commands, refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and *Cisco IOS Wide-Area Networking Command Reference.*

VoFR provides support for various FRF.11 features depending on the hardware platform used (see Table 30).

Table 30 FRF.11 Forum Features Supported by Hardware Platform

FRF.11 Forum Features	Cisco MC3810 Multiservice Concentrator	Cisco 2600/3600 Series Routers	Cisco 7500 Series Routers with VIP Support
Class 1–Compliance Requirements (sec. 4.1)	Not supported	Not supported	Not supported
Class 2–Compliance Requirements (sec. 4.2)	Supported	Supported	Supported
Annex A–Dialed Digits Transfer Syntax	Supported	Supported	Supported
Annex B–Signaling Bit Transfer Syntax	Supported	Supported	Supported
Annex C–Data Transfer Syntax	Supported	Supported	Supported
Annex D–Fax Relay Transfer Syntax	Supported	Supported	Supported
Annex E–CS-ACELP Transfer Syntax (G.729/G.729A)			
• Sequence Number	Supported	Supported	Supported
• Packing Factor	Supported	Supported	Supported
Annex F–Generic PCM/ADPCM Voice Transfer Syntax	Supported	Supported	Supported
Annex G –G.727 Discard-Eligible E-ADPCM Voice Transfer Syntax	Not supported	Not supported	Not supported
Annex H–G.728 LD-CELP Transfer Syntax	Not supported	Supported	Supported
Annex I–G.723.1 Dual Rate Speech Coder	Not supported	Supported	Supported
Transmission and reception of multiple subframes within a single Frame Relay frame	Not supported	Not supported	Not supported

End-to-End FRF.12 Fragmentation

FRF.12 fragmentation is defined by the FRF.12 standard. The FRF.12 implementation agreement enables long data frames to be fragmented into smaller pieces and interleaved with real-time frames. In this way, real-time voice and nonreal-time data frames can be carried together on lower-speed links without causing excessive delay to the real-time traffic.

Use this fragmentation type when the PVC is not carrying voice, but is sharing the link with other PVCs that are carrying voice. The fragmentation header is included only for frames that are greater than the fragment size configured. FRF.12 is the recommended fragmentation for VoIP packets.

Note VoIP packets should not be fragmented. However, VoIP packets can be interleaved with fragmented packets.

The Cisco 2600 series, 3600 series, and 7200 series routers and the Cisco MC3810 multiservice concentrator support end-to-end fragmentation on a per-PVC basis. Fragmentation is configured through a map class that applies to one or many PVCs, depending on how the class is applied.

When end-to-end FRF.12 fragmentation is used, the VoIP packets do not include the FRF.12 header, provided the size of the VoIP packet is smaller than the fragment size configured. However, when FRF.11 Annex C or Cisco proprietary fragmentations are used, VoIP packets do include the fragmentation header.

Frame Relay Fragmentation Using FRF.11 Annex C

When VoFR and fragmentation are configured on a PVC, the Frame Relay fragments are sent in the FRF.11 Annex C format. FRF.11 fragmentation is used when voice traffic is sent on the PVC, and Annex C format is used for data. With FRF.11, all data packets contain fragmentation headers, regardless of size. This form of fragmentation is not recommended for use with VoIP.

Cisco Proprietary Voice Encapsulation

Cisco proprietary voice encapsulation was implemented for the Cisco MC3810 multiservice concentrator and was used for data packets on a PVC and voice traffic. This fragmentation type is used on data packets on PVCs that carry voice traffic.

When VoFR is configured on a DLCI and fragmentation is enabled on a map class, the Cisco 7500 series router with Versatile Interface Processor (VIP) can interoperate with Cisco 2600 series, 3600 series, 7200 series, and other 7500 series routers as tandem nodes, but it cannot perform call termination with Cisco MC3810 multiservice concentrators running Cisco IOS releases *before* 12.0(3)XG or 12.0(4)T.

Map Classes and Voice Packet Queues

You must create and configure a Frame Relay map class before configuring a Frame Relay DLCI for voice traffic. The map class has configuration information about voice bandwidth, fragmentation size, and traffic shaping attributes. These attributes are required for sending voice traffic on the PVC.

Traffic Shaping

When a Frame Relay PVC is configured to support voice traffic, the carrier must be able to accommodate the traffic rate or profile sent on the PVC. If too much traffic is sent at once, the carrier might discard frames causing disruptions to real-time voice traffic. The carrier might also deal with traffic bursts by queueing up the bursts and delivering them at a metered rate. Excessive queueing also causes disruption to real-time voice traffic. Traffic shaping compensates for this condition and is necessary to prevent the carrier from discarding eligible discard bits on ingress and to prevent excessive burst data from affecting voice quality.

When the outgoing Excess Burst (Be) size is configured, the Committed Burst (Bc) size and the committed information rate (CIR) values must be obtained from the carrier. The configured values on the router must match those of the carrier.

VoFR Prerequisite Tasks

Before configuring the router for VoFR, perform the following tasks:

- Complete the company dial plan and establish a working telephony network based on the dial plan:
  - Integrate the dial plan and telephony network into the existing Frame Relay network topology. Make routing or dialing transparent to the user; for example, avoid secondary dial tones from secondary switches, where possible.
  - Contact the PBX vendor for instructions on how to reconfigure the appropriate PBX interfaces.
- Establish a working IP and Frame Relay network. For more information about configuring IP, see the "IP Overview," "Configuring IP Addressing," and "Configuring IP Services" chapters in the *Cisco IOS IP Configuration Guide*. For more information about configuring Frame Relay, see the *Cisco IOS Wide-Area Networking Configuration Guide*.
- Configure the required codecs and POTs dial peer configurations in "Configuring Dial Peers, Dial Plans, and Digit Manipulation" chapter.
- Configure voice ports. For more information, see the "Configuring Voice Ports" chapter.
- Configure the clock source interfaces. For more information, refer to the "Configuring Synchronous Clocking" appendix.

VoFR Configuration Task List

This section describes the following tasks:

For information regarding the configuring of voice ports and dial peers, refer to the "Configuring Voice Ports" and "Configuring Voice Dial Peers, Dial Plans, and Digit Manipulation" chapters.

Configuring Frame Relay to Support Voice

To configure Frame Relay to support voice, a map class must be applied to a single DLCI or to a group of DLCIs, depending on how the class has been applied to the virtual circuit. If there is a large number of PVCs to configure, assign the same traffic-shaping properties to the PVCs. The values for each PVC are not statically defined. Multiple map classes with different variables for each map class can also be created.

When the **frame-relay voice bandwidth** command is entered, a special queue is created for voice packets only so that time-sensitive voice packets have preference over data packets.

This section describes the configuration of map classes as follows:

- Configuring a Map Class to Support Voice Traffic, page 430
- Configuring a Map Class for Traffic-Shaping Parameters, page 431

To configure the map class to support FRF.12 fragmentation, refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and *Command Reference* for more information.

Configuring a Map Class to Support Voice Traffic

When you are configuring a Frame Relay map class to support voice traffic, you must reserve the appropriate amount of voice bandwidth. If there is not enough bandwidth reserved, new calls are rejected. When calculating the amount of required voice bandwidth, include the voice packetization overhead and not just the raw compressed speech codec bandwidth.

Remember that there are a six or seven bytes of total overhead per voice packet, including standard Frame Relay headers and flags. For subchannels (CIDs) numbered less than 64, the overhead is 6 bytes. For subchannels numbered greater than or equal to 64, the overhead is 7 bytes. Add one byte if voice sequence numbers are enabled in the voice packets.

To determine the required voice bandwidth, use the following calculation:

required_bandwidth = codec_bandwidth * (1 + overhead/payload_size)

This calculation addresses the amount of bandwidth consumed on the physical network interface. The figure does not necessarily represent the amount of connection bandwidth used within the Frame Relay network itself, which may be higher because the overhead of switching small packets.

When 30-ms duration voice packets are used, an approximate general rule is to add 2000 bps overhead to the raw voice compressed speech codec rate. With the 32 kbps G.726 adaptive differential pulse code modulation (ADPCM) speech coder, a 30-ms speech frame uses 120 bytes voice payload plus 6 to 7 bytes overhead, and the overall bandwidth requirement is about 34 kbps for each call.

To configure a Frame Relay map class to support voice traffic on DLCIs, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# map-class frame-relay map-class-name`	Creates a map class name to assign to a group of PVCs and enters map-class configuration mode. A map class name must be unique.
Step 2	`Router(config-map-class)# frame-relay voice bandwidth bps_reserved [queue depth]`	Enters the bandwidth in bits per second (bps) and determines the number of voice calls enabled on the DLCIs where the map class is associated. The keywords and arguments are as follows: • *bps_reserved*—Reserved bandwidth. Valid range is from 8,000 to 45,000,000 bps. The default is 0 (disables all voice calls). • **queue** *depth*—(Optional) Specifies the size of the voice queue. Valid range is from 30 to 100. The default is 100.

Note It is recommended that the bps be no higher than the minimum CIR if the voice quality is impacted when burst is being sent.

Configuring a Map Class for Traffic-Shaping Parameters

To configure a Frame Relay map class for the traffic shaping parameters for one or more DLCIs, use the following commands in map-class configuration mode:

	Command	Purpose
Step 1	`Router(config-map-class)# frame-relay bc out` *bits*	Configures the outgoing bc size for this group of PVCs. Configure the *bits* value to a minimum of 1000 for voice traffic. Ensure that the bc size matches the carrier to prevent the carrier from discarding DE bits on ingress.
Step 2	`Router(config-map-class)# frame-relay be out` *bits*	Configures the outgoing be size for this group of PVCs. Ensure that the Excess Burst size matches the carrier to prevent the carrier from discarding DE bits on ingress.
Step 3	`Router(config-map-class)# frame-relay min-cir {in | out}` *bps*	Configures the minimum acceptable incoming or outgoing CIR for this group of PVCs.
Step 4	`Router(config-map-class)# frame-relay cir out` *bits*	Configures the outgoing excess CIR for this group of PVCs. Configured the CIR size to match your carrier to prevent the carrier from discarding DE bits on ingress.
Step 5	`Router(config-map-class)# frame-relay cir in` *bits*	(Optional) Configures the incoming CIR size for this group of PVCs.
Step 6	`Router(config-map-class)# frame-relay adaptive shaping becn`	(Optional) Configures the adaptive traffic rate adjustment to support backward explicit congestion notification (BECN) on this group of PVCs.

Configuring VoFR Dial Peers

To configure a VoFR dial peer, you must uniquely identify the peer (by assigning it a unique tag number) and define the outgoing serial port number and the virtual circuit number.

Depending on your dial plan configuration, you might need to consider how to configure voice networks with variable-length dial plans, number expansion, excess digit playout, forward digits, and default voice routes, or use hunt groups with dial peer preferences.

Note On the Cisco MC3810 multiservice concentrator, a voice class can be configured to assign idle state and out-of-service (OOS) signaling attributes to a VoFR dial peer. For more information, see the "Configuring Trunk Connections and Conditioning Features" chapter.

To configure a VoFR dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `vofr`	Defines a VoFR dial peer and enters dial peer configuration mode. All subsequent commands that are entered in dial peer voice configuration mode before exiting apply to this dial peer. The *number* argument identifies the dial peer and must be unique on the router. Do not duplicate a specific tag number.
Step 2	`Router(config-dial-peer)#` **destination-pattern**`[+]`*string*`[T]`	Configures the dial peer destination pattern. The same restrictions for the string listed in the POTS dial peer configuration also apply to the VoFR destination pattern. Also configures standard VoFR dial peers for switched calls on the tandem routers. • Plus sign (**+**)—(Optional) Indicates an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Specifies the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – Asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. On the Cisco 3600 only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,) inserts a pause between digits. – Period (.) matches any entered digit (this character is used as a wildcard). On the Cisco 3600, the period cannot be used as a leading character in a string (for example, .650). • **T**—(Optional) Indicates that the destination-pattern value is a variable length dial-string. **Note** Tandem-switched calls are not allowed when the call type is an FRF.11 trunk call. The Cisco 7200 series routers can serve only as tandem nodes in the VoFR network using Cisco IOS Release 12.1. This is the only dial peer procedure supported on the Cisco 7200 series.

	Command	Purpose
Step 3	`Router(config-dial-peer)# session target interface dlci [cid]`	Configures the Frame Relay session target for the dial peer. **Note** The *cid* argument is required for FRF.11 trunk calls.
Step 4	`Router(config-dial-peer)# session protocol {cisco-switched | frf11-trunk}`	(Optional) Configures the session protocol to support switched calls or FRF.11 trunk calls. If FRF.11 trunk calls are sent over the Frame Relay network, the VoFR dial peers must be statically configured on both sides of the trunk specifically to support FRF.11 trunk calls. FRF.11 trunk calls cannot be used in conjunction with dial plans or be sent through tandem nodes. **Note** The **cisco-switched** keyword is the default.
Step 5	`Router(config-dial-peer)# codec {type} [bytes payload_size]`	Specifies the voice coder rate of speech and payload size for the dial peer. The default dial peer codec is **g729r8**. The keywords and arguments are as follows: • *type*—Specifies the coder rate of speech. The rates are hardware-specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. • **bytes**—(Optional) Specifies the payload size. Each codec type defaults to a different payload size if a value is not specified. • *payload_size*—(Optional) Specifies the payload size by entering the **bytes** value. Each codec type defaults to a different payload size if a value is not specified. To obtain a list of the default payload sizes, enter the **codec** command and the **bytes** option followed by a question mark (?). **Note** The Cisco MC3810 multiservice concentrator is limited to a maximum of 12 calls when using **g729r8**. Use **g729ar8** to support up to 24 calls on the Cisco MC3810 multiservice concentrator. **Note** If configuring switched voice calls on the Cisco MC3810 multiservice concentrator, configure the codec type on the voice port. **Note** For FRF.11 trunk calls, the **codec** values must be set the same on both sides of the connection.

	Command	Purpose
Step 6	`Router(config-dial-peer)# dtmf-relay`	(Optional) Specifies support for the DTMF relay to improve end-to-end transport of the DTMF tones, if the codec type configured is a low bit-rate codec such as **g729** or **g723**. DTMF tones do not always propagate reliably with low bit-rate codecs. DTMF relay is disabled by default.
Step 7	`Router(config-dial-peer)# signal-type {cas | cept | ext-signal | transparent}`	If Cisco trunk permanent calls are being configured, the signal type is required. The signal type defines the ABCD signaling packets that are generated by the voice port and sent to the data network. Use the **cas**, **cept**, **ext-signal**, and **transparent** keywords. To configure FRF.11 calls, use only the **cas** and **ext-signal** keywords. These keywords are optional on Cisco 2600/3600 series routers and configure the signal type on these routers for FXS-FXS trunks. The keywords are as follows: • **cas**—Default signaling type that is North American CAS/robbed-bit signaling. • **cept**—Provides basic E1 ABCD protocol, primarily Conférence Européenne des Postes et des Télécommunications (CEPT) E&M signaling, on the Cisco MC3810 multiservice concentrator. This keyword is used for European voice networks. If the keyword is used with FXS or FXO voice ports, the signaling is equivalent to Mercury Exchange Limited (MEL) CAS. The keyword is not supported on the Cisco 2600/3600 series. • **ext-signal**—Used for required external signaling channels (for example, common channeling signaling), or when no signaling information is sent over a permanent “dumb” voice pipe (for example, carrying audio for a public address system).

	Command	Purpose
		• **transparent**—Used on the Cisco MC3810 multiservice concentrator with *digital* voice ports when the ABCD signaling bits are copied and passed transparently from the T1/E1 interface without interpretation (also known as transparent FRF.11 signaling). The keyword enables the Cisco MC3810 multiservice concentrator to handle or transport unknown signaling protocols. On the Cisco MC3810 multiservice concentrator with *analog* voice ports, the **transparent** keyword does not apply and is equivalent to the **cept** keyword. This keyword is not supported on the Cisco 2600 series and 3600 series in Cisco IOS Release 12.2. **Note** By default, the Cisco MC3810 multiservice concentrator, when configured using **transparent**, operates the voice path in a permanently open state so that voice packets are sent (and network bandwidth consumed) regardless of the state of the call. The signal type must be configured in such a way that the signal type is the same at both ends of the permanent voice call. When a permanent connection is configured between a T1/E1 Cisco MC3810 multiservice concentrator and an analog voice port on a Cisco 2600 or Cisco 3600 series routers, the signal type should be set to **cas**, which is the default.
Step 8	`Router(config-dial-peer)# called-number` *termination-string*	Required for the Cisco 2600/3600 series routers only. Configures the termination string for FRF.11 trunk calls. This command is required to enable the router to establish an incoming trunk connection. This command applies only when the **session protocol** command is set to **frf11-trunk**. **Note** Although this command is visible on the Cisco MC3810 multiservice concentrator, the command is disabled.
Step 9	`Router(config-dial-peer)# no vad`	(Optional) Disables VAD on the dial peer. This command is enabled by default.
Step 10	`Router(config-dial-peer)# sequence-numbers`	(Optional) Enables the voice sequence number if required for your configuration. This command is disabled by default.

	Command	Purpose
Step 11	`Router(config-dial-peer)# preference value`	(Optional) Configures a preference for the VoFR dial peer. The *value* argument is a number from 0 to 10 where the lower the number, the higher the preference in hunt groups.
Step 12	`Router(config-dial-peer)# fax rate {2400 | 4800 | 7200 | 9600 | 14400 | disable | voice}`	(Optional) Configures the transmission speed (in bps) at which a fax will be sent to the dial peer. The default is **voice**, which specifies the highest possible transmission speed allowed by the voice rate.

To configure another VoFR dial peer, exit dial peer configuration mode and repeat Steps 1 through 10.

Note Repeat this procedure on the destination router on the other side of the FRF.11 trunk.

Configuring Switched Calls

To configure switched calls on Cisco 2600, 3600, and 7200 series routers and Cisco MC3810 multiservice concentrators, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# frame-relay interface-dlci dlci`	Enters the DLCI configuration mode.
Step 2	`Router(config-fr-dlci)# vofr [data cid] [call-control] [cid]`	Configures the Frame Relay DLCI to support VoFR. When the **vofr** command is used, all subchannels on the DLCI are configured for FRF.11 encapsulation The keywords and arguments are as follows: • **data**—Selects a subchannel (CID) for data other than the default subchannel (CID 4). The recommended setting is **vofr data** *4* **call-control** *5*. • *cid*—Specifies the subchannel to use for data. Valid values are from 4 to 255. The default is 4. If data is specified, a valid CID must be entered. • **call-control**—(Optional) Specifies that a subchannel is reserved for call-control signaling. Call-control is not supported on Cisco MC3810 multiservice concentrators. • *cid*—(Optional) Specifies the subchannel to use for call-control signaling. Valid values are from 4 to 255. The default is 5. If call-control is specified and a CID is not entered, the default CID is used. If the **vofr** command is entered without any keywords or arguments, the data subchannel (*cid*) is 4 and there is no call-control subchannel. **Note** The **vofr** command uses WFQ at the PVC level. If the **vofr cisco** command is used, WFQ cannot be disabled.
	or `Router(config-fr-dlci)# vofr cisco`	Configures the DLCI and the Cisco proprietary voice encapsulation for switched calls to Cisco MC3810 multiservice concentrators. When this command is entered, data CID 4 and call-control CID 5 are automatically assigned. If user-dialed calls are being configured, stop here. If auto-ringdown calls are being configured, continue to the next step.
Step 3	`Router(config)# voice-port`	Identifies the voice port to configure and enters the voice-port configuration mode. **Note** The **voice-port** command is hardware specific. For more information, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.
Step 4	`Router(config-voice-port)# connection [plar | tie-line] destination-string`	Configures the private-line, auto-ringdown (PLAR) or tie-line connection, specifying the telephone number in the *destination-string*.

Table 31 lists the supported VoFR connections and the appropriate commands to configure switched calls.

Table 31 Supported VoFR Connections for Switched Calls

Switched Calls (User-Dialed or Auto-Dialed)	Data Fragmentation Supported	Frame Relay DLCI Command[1]	Session Protocol Command[2]	Voice Port Command
To routers supporting VoFR	FRF.11 Annex C	**vofr** [**data** *cid*] [**call-control** [*cid*]][3]	**session protocol cisco-switched**[4]	For user-dialed calls: none For auto-ringdown calls: **connection plar** *destination-string*
To a Cisco MC3810 multiservice concentrator running Cisco IOS Releases before 12.1(2)T	Cisco proprietary[5]	**vofr cisco**[6]	**session protocol cisco-switched**	For user-dialed calls: none For auto-ringdown calls: **connection plar** *destination-string*

1. The **voice-encap** option of the **frame-relay interface-dlci** command on the Cisco MC3810 multiservice concentrator is no longer supported.
2. Dial peer configuration mode.
3. The recommended use of this command is **vofr data 4 call-control 5**.
4. The **session protocol cisco-switched** command is the default setting. If the command is not entered, the setting still applies.
5. Cisco proprietary fragmentation is based on an early draft of FRF.12 and is compatible with Cisco MC3810 multiservice concentrators.
6. This command uses data CID 4 and call-control CID 5.

Tandem Switching of Switched Calls

Depending on which router is the end node and which is the tandem node, the correct Frame Relay PVC type must be configured. Table 32 shows the router combinations that can serve as end and tandem nodes and the command that is required to enable VoFR.

Table 32 VoFR End and Tandem Node Combinations

End Node	Tandem Node	Required VoFR Command
Cisco 2600, Cisco 3600, or Cisco 7200 and Cisco MC3810 multiservice concentrator	Cisco 2600, Cisco 3600, or Cisco 7200 and Cisco MC3810 multiservice concentrator	**vofr call-control**
Cisco 2600 or Cisco 3600 and Cisco MC3810 multiservice concentrator	Cisco MC3810 multiservice concentrator running Cisco IOS releases before 12.1(2)T	**vofr cisco**
Cisco MC3810 multiservice concentrator running Cisco IOS releases before 12.1(2)T	Cisco 2600, Cisco 3600, or Cisco 7200	**vofr cisco**

Note When you are creating voice networks with a mixture of router types, the Cisco MC3810 multiservice concentrator must be running Cisco IOS Release 12.0(3)XG, 12.0(4)T, or later releases, to act as a tandem node. For each configured tandem node, two VoFR dial peers must be configured, one for each tandem connection.

To configure VoFR dial peers on tandem routers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number vofr`	Defines a VoFR dial peer and enters dial peer configuration mode. All subsequent commands that are entered in dial peer voice configuration mode before exiting apply to this dial peer.
Step 2	`Router(config-dial-peer)# destination-pattern [+]string[T]`	Configures the dial peer destination pattern. The same restrictions for the string listed in the POTS dial peer configuration also apply to the VoFR destination pattern.
Step 3	`Router(config-dial-peer)# session target interface dlci`	Configures the Frame Relay session target for the dial peer.
Step 4	`Router(config-dial-peer)# preference value`	(Optional) Configures a preference for the VoFR dial peer. The *value* argument is a number from 0 to 10 where the lower the number, the higher the preference in hunt groups.

To configure the next VoFR dial peer, exit dial peer configuration mode by entering **exit**, and repeat Steps 1 through 4. On tandem nodes, at least two VoFR dial peers are required, one for each call leg.

Configuring Cisco Trunk Calls

Before configuring the Cisco trunk calls, consider the following restrictions and recommendations:

- VoFR dial peers must be configured to send Cisco trunk calls over the Frame Relay network. Cisco trunk calls are permanent calls. One critical task is configuring the signal type for the dial peer. It must be the same at both ends of the permanent voice call. See the "Configuring Dial Peers, Dial Plans, and Digit Manipulation" chapter for more information.
- When a permanent connection between a T1/E1 Cisco MC3810 multiservice concentrator and an analog voice port on a Cisco 2600 or Cisco 3600 series routers is configured, the default signal type is **cas**.
- Use of Cisco trunks for permanent calls is recommended over FRF.11 trunk calls unless FRF.11 compliant standards-based interworking is required with non-Cisco devices. The Cisco trunk protocol is a superset of the FRF.11 protocol and contains Cisco proprietary extensions designed to support switched call routing and other advanced features.

Table 33 lists the supported VoFR connections and the commands to enter.

Table 33 VoFR Connections for Cisco Trunk Calls

Cisco Trunk Calls	Data Fragmentation Supported	VoFR Command	Session Protocol Command[1]	Voice Port Command
To routers supporting VoFR	FRF.11 Annex C	**vofr data** *cid* **call-control** *cid*	**session protocol cisco-switched**	**connection trunk** *destination-string* [**answer mode**]
To a Cisco MC3810 multiservice concentrator running Cisco IOS Releases before 12.0(7) XK and 12.1(2)T	Cisco proprietary	**vofr cisco**[2]	**session protocol cisco-switched**	**connection trunk** *destination-string* [**answer mode**]

1. The **session protocol cisco-switched** command, whether entered or not, is the default setting.
2. When the **cisco** keyword is entered, Cisco proprietary data implementation is enabled. This implementation is used only for backward compatibility to earlier releases.

To configure Cisco trunk permanent calls, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# frame-relay interface-dlci dlci`	Configures the DLCI to support VoFR. **Note** The **voice-encap** option of the **frame-relay interface-dlci** command on the Cisco MC3810 multiservice concentrator is no longer supported beginning in Cisco IOS 12.2.
Step 2	`Router(config-if)# vofr [[cisco] | [[data cid] [call-control][cid]]]`	Enables VoFR on the DLCI. If the **vofr** command is entered without any keywords or arguments, the data subchannel is CID 4, and there is no call-control subchannel. **Note** When the **vofr** command is used, all subchannels on the DLCI are configured for FRF.11 encapsulation. This configuration uses the standard FRF.11 Annex C fragmentation. The **vofr** command uses WFQ at the PVC level. If the **vofr cisco** command is used, WFQ cannot be disabled. If only tandem calls are being configured, stop here, otherwise proceed to Step 3.
Step 3	`Router(config]# voice-port`	Identifies the voice port to configure and enters voice-port configuration mode. **Note** The **voice-port** command is hardware specific. See the *Cisco IOS Voice, Video, and Fax Command Reference Guide* for more information.
Step 4	`Router(config-voice-port)# connection trunk destination-string [answer-mode]`	Configures the trunk connection by specifying the telephone number in *destination-string*. One side must be the call initiator (master) and the other side is the call answerer (slave). By default, the voice port is the master. The **answer-mode** keyword specifies the voice port that operates in slave mode.
Step 5	`Router(config-voice-port)# shutdown`	Shuts down the voice port.
Step 6	`Router(config-voice-port)# no shutdown`	Reactivates the voice port to enable the trunk connection.

Note When the **connection trunk** or **no connection trunk** command is entered, the voice port must be toggled by entering **shutdown**, and then **no shutdown** before the changes take effect.

Configuring FRF.11 Trunk Calls

On the Cisco MC3810 multiservice concentrators and Cisco 2600 and 3600 series routers, FRF.11 trunk calls to a second router can be configured, except tandem FRF.11 trunk calls. Configuring FRF.11 trunk calls to a second router requires that the **session protocol** dial peer configuration command be set to **frf11-trunk**.

Table 34 lists the supported VoFR connections and the required commands to configure FRF.11 trunk calls.

Table 34 VoFR Connections for FRF.11 Trunk (Private-Line) Calls

FRF.11 Trunk Calls	Data Fragmentation Supported	VoFR DLCI Command[1]	Session Protocol Command	Voice Port Command
To routers supporting VoFR	FRF.11 Annex C	**vofr** [**data** *cid*] [**call-control** *cid*][2]	**session protocol frf11-trunk**	**connection trunk** *destination-string* [**answer mode**]

1. Dial peer configuration mode.
2. For FRF.11 trunk calls, the call-control option is not required. It is required only if you mix FRF.11 trunk calls with other types of voice calls on the same PVC.

To configure FRF.11 trunk calls, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# frame-relay interface-dlci dlci`	Configures the DLCI and enters DLCI configuration mode.
Step 2	`Router(config-fr-dlci)# vofr [data cid] [call-control cid]`	Configures the DLCI and optionally enters the data and call-control CIDs. When the keywords and arguments are configured, all subchannels on the DLCI are configured for FRF.11 encapsulation except the data subchannel. If no keywords or arguments are entered, the data subchannel is CID 4, and there is no call-control subchannel.
Step 3	`Router(config)# voice-port`	Identifies the voice port to configure and enters voice-port configuration mode. **Note** The **voice-port** command is hardware specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference* publication for more information.
Step 4	`Router(config-voice-port)# connection trunk destination-string [answer-mode]`	Configures the trunk connection by specifying the telephone number in *destination-string*. One side of a call must act as the call initiator (master) and the other side as the call answerer (slave). By default, the voice port is the master.
Step 5	`Router(config-voice-port)# shutdown`	Shuts down the voice port.
Step 6	`Router(config-voice-port)# no shutdown`	Reactivates the voice port to enable the trunk connection.

Note When the **connection trunk** or **no connection trunk** command is entered, the voice port must be toggled by entering **shutdown**, and then **no shutdown** before the changes take effect.

Verifying the Voice Connections

To verify switched calls voice connections, perform the following tasks:

- Pick up the telephone handset and verify that there is a dial tone.
- Call from a local telephone to the configured dial peer and verify that the call completes.

To verify the FXO-FXS trunk calls to a remote PBX, perform the following tasks:

- Pick up the telephone and listen for a dial tone from the remote PBX.
- Dial a telephone number, so that the remote PBX routes the call.

To verify the voice connections, perform the following tasks:

- Check the validity of the dial peer and voice port configuration by performing the following tasks:
  - Enter the **show dial-peer voice** command to verify that the data configured is correct.
  - Enter the **show dial-peer voice summary** command to check the validity of the dial peer configurations.
  - Enter the **show voice port** command to show the status of the voice ports.
  - Enter the **show call active voice** with the keyword **brief** to show the call status for all voice ports.
  - Enter the **show voice call** command to check the validity of the voice port configuration.
  - Enter the **show voice dsp** command to show the current status of all DSP voice channels.
  - Enter the **show voice permanent** command to show the status of Cisco trunk permanent calls.
  - Enter the **show call history** command to show the active call table.
- Check the validity of the VoFR configuration on the DLCI by performing the following task:
  - Enter the **show frame-relay vofr** [*interface* [*dlci* [*cid*]]] command to show the VoFR configuration. This command is not supported on the Cisco MC3810 multiservice concentrator when the **vofr cisco** command is configured.

Verifying the Frame Relay Configuration

Check the validity of the configuration by performing the following tasks:

- Enter the **show frame-relay pvc** command to show the status of the PVCs.
- Enter the **show frame-relay vofr** command with the arguments *interface*, *dlci, and cid* to show statistics and information on the open subchannels. This command does not display if the **vofr cisco** command is entered on the Cisco MC3810 multiservice concentrator.
- Enter the **show frame-relay fragment** command with the arguments *interface number* and *dlci* to show the Frame Relay fragmentation configuration.
- Enter the **show traffic-shape queue** command to display the traffic-shaping information if Frame Relay traffic shaping is configured. The **queue** option displays the queueing statistics.

Troubleshooting Tips

To troubleshoot and resolve configuration issues, perform the following tasks:

- If no calls are going through, ensure that the **frame-relay voice bandwidth** command is configured.
- If VoFR is configured on a PVC and there are problems with data connectivity on that PVC, ensure that the **frame-relay fragment** command has been configured.
- If data is not being transmitted but fragmentation is configured, ensure that Frame Relay traffic shaping is turned on.
- If the problem is with the dial plan or the dial peers, use the **show dial-plan number** command with the argument *dial string* to display which dial peers are being used when a specific number is called.
- If there are problems connecting an FRF.11 trunk call, ensure that the **session protocol** dial peer command is set to **frf11-trunk**.
- If FRF.11 trunk calls on the Cisco 2600 or Cisco 3600 series routers are being configured, verify that the **called-number vofr** dial peer command is configured and that its number matches the destination pattern of the corresponding POTS dial peer.
- Ensure that the voice port is set to **no shutdown**.
- Ensure that the serial port or the T1/E1 controller is set to **no shutdown**.
- Toggle the voice port by first entering **shutdown**, and then **no shutdown** every time the **connection trunk** or **no connection trunk** command is entered.

Monitoring and Maintaining the VoFR Configuration

To monitor and maintain the VoFR configuration, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# show call active voice [brief]`	Displays the active call table.

`Router# show call history voice [last number] | [brief]`
or
`Router# show call history voice record`
	Displays the call history table.

`Router# show dial-peer voice`	Displays configuration information and call statistics for dial peers.
`Router# show frame-relay fragment`	Displays information about the Frame Relay fragmentation taking place in the Cisco router.
`Router# show frame-relay pvc`	Displays statistics about PVCs for Frame Relay interfaces.
`Router# show frame-relay vofr`	Displays the FRF.11 subchannels information on VoFR DLCIs.
`Router# show interfaces serial`	Displays information about a serial interface.
`Router# show traffic-shape queue`	Displays information about the elements queued at a particular time at the VC (DLCI) level.

Command	Purpose
`Router# show voice call`	Displays the call status for all voice ports on the Cisco MC3810 multiservice concentrators.
`Router# show voice permanent-call`	Displays information about the permanent calls on a voice interface.

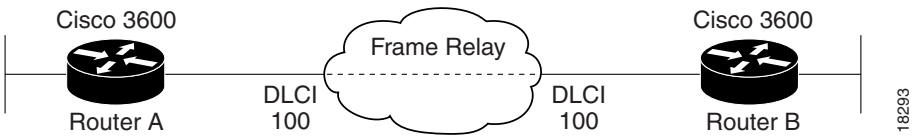
VoFR Configuration Examples

This section provides specific configuration examples for VoFR connections and includes:

Two Routers Using Frame Relay Fragmentation Example

Figure 92 shows an example of Frame Relay fragmentation between two routers. This configuration uses FRF.12 fragmentation.

Figure 92 Two Routers Using Frame Relay Fragmentation



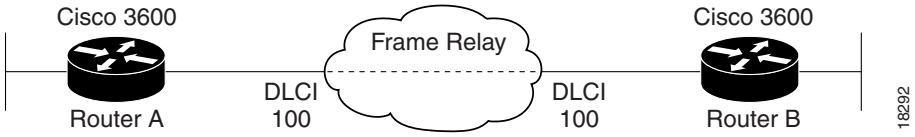
Router A	Router B

`hostname 3600A`
`!`
`interface serial 0/0`
`ip address xxx.xxx.xxx 255.255.255.0`
`frame-relay traffic shaping`
`!`
`frame-relay interface-dlci 100`
`class toto`
`!`
`map-class frame-relay toto`
`encapsulation frame-relay`
`frame-relay cir s`
`frame-relay bc u`
`frame-relay fragment y`
	`hostname 3600B`
	`!`
	`interface serial 0/0`
	`ip address xxx.xxx.xxx 255.255.255.0`
	`frame-relay traffic shaping`
	`frame-relay class toto`
	`frame-relay interface-dlci 100`
	`!`
	`map-class frame-relay toto`
	`encapsulation frame-relay`
	`frame-relay cir s`
	`frame-relay bc u`
	`frame-relay fragment y`

Two Routers Using a VoFR PVC Example

Figure 93 shows an example of two routers that use FRF.11 Annex C fragmentation with connections using a VoFR PVC.

Figure 93 Two Cisco 3600 Series Routers Using a VoFR PVC



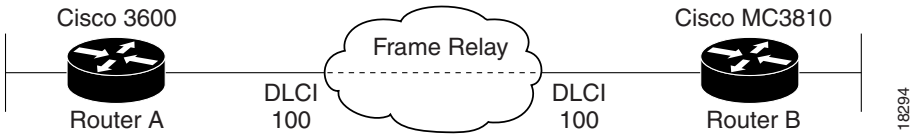
Router A	Router B

`hostname 3600A`
`!`
`interface serial 0/0`
`frame-relay traffic shaping`
`!`
`frame-relay interface-dlci 100`
`vofr data z`
`class toto`
`!`
`map-class frame-relay toto`
`frame-relay voice-bandwidth t`
`frame-relay min-cir x`
`frame-relay cir s`
`frame-relay bc u`
`frame-relay fragment y`
	`hostname 3600B`
	`!`
	`interface serial 0/0`
	`frame-relay traffic shaping`
	`frame-relay class toto`
	`!`
	`frame-relay interface-dlci 100`
	`vofr data z`
	`!`
	`map-class frame-relay toto`
	`frame-relay voice-bandwidth t`
	`frame-relay min-cir x`
	`frame-relay cir s`
	`frame-relay bc u`
	`frame-relay fragment y`

Router Using VoFR PVCs Connected to Cisco MC3810s Before 12.1(2)T Example

Figure 94 shows an example of a Cisco 3600 series router with connections to a Cisco MC3810 multiservice concentrator running a Cisco IOS release before12.1(2)T. In this example, the VoFR interface on both the Cisco 3600 series router and the Cisco MC3810 multiservice concentrator is configured by using the **vofr cisco** command. This configuration uses FRF.11 Annex C fragmentation.

Figure 94 Router Using VoFR PVCs Connected to a Cisco MC3810 Multiservice Concentrator

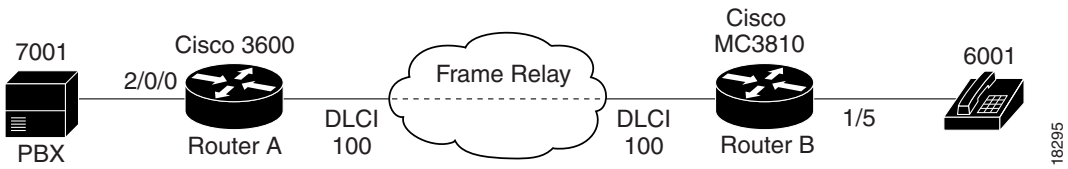


Router A	Router B
interface serial 0/0	interface serial 0
ip address xxx.xxx.xxx	ip address xxx.xxx.xxx
255.255.255.0	255.255.255.0
frame-relay traffic shaping	frame-relay traffic shaping
!	frame-relay class *toto*
frame-relay interface-dlci 100	!
vofr cisco	frame-relay interface-dlci 100
class toto	vofr cisco
!	!
map-class frame-relay toto	map-class frame-relay toto
frame-relay voice-bandwidth t	frame-relay voice-bandwidth t
frame-relay min-cir x	frame-relay min-cir x
frame-relay cir s	frame-relay cir s
frame-relay bc u	frame-relay bc u
frame-relay fragment y	frame-relay fragment y

Cisco Trunk Calls Between Two Routers Example

Figure 95 shows an example of VoFR Cisco trunk calls between two routers.

Figure 95 Cisco Trunk (Private-Line) Calls Between Two Routers



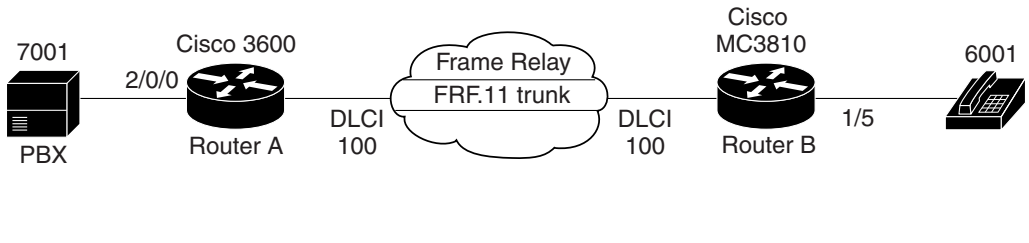
Router A	Router B
interface serial 0/0	interface serial 0
ip address xxx.xxx.xxx	ip address xxx.xxx.xxx
255.255.255.0	255.255.255.0
encapsulation frame-relay	encapsulation frame-relay
frame-relay traffic shaping	frame-relay traffic shaping
frame-relay interface-dlci 100	frame-relay interface-dlci 100
class voice	class voice
vofr data 4 call-control 5	vofr data 4 call-control 5
!	!
map-class frame-relay voice	map-class frame-relay voice
frame relay cir s	frame relay cir s
frame relay bc u	frame relay bc u
frame-relay voice bandwidth v	*frame-relay voice bandwidth v*
frame-relay min-cir x	frame-relay min-cir x
frame-relay fragment y	frame-relay fragment y
!	!
voice-port 2/0/0	voice-port 1/5
connection trunk 6001 answer-mode	connection trunk 7001
!	!
dial-peer voice 1 pots	dial-peer voice 2 pots
destination-pattern 7001	destination-pattern 6001

Router A	Router B
port 2/0/0	port 1/5
!	!
dial-peer voice 2 vofr	dial-peer voice 4 vofr
codec x bytes y	codec x bytes y
destination-pattern 6001	destination-pattern 7001
session protocol cisco-switched	session protocol cisco-switched
session target Sn 100	session target Sn 100

FRF.11 Trunk Calls Between Two Routers Example

Figure 96 shows an example of FRF.11 trunk calls configured between two routers.

Figure 96 FRF.11 Trunk Calls Between Two Routers

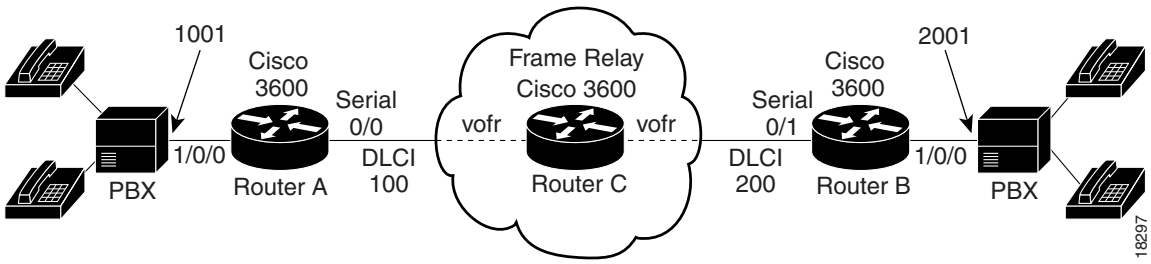


Router A	Router B
hostname 3600A ! interface serial 0/0 ip address xxx.xxx.xxx 255.255.255.0 encapsulation frame-relay frame-relay traffic shaping frame-relay interface-dlci 100 class voice vofr data 4 ! map-class frame-relay voice frame-relay cir s frame-relay min-cir in x frame-relay bc u frame-relay voice bandwidth v frame-relay fragment y ! voice-port 2/0/0 connection trunk 6001 ! dial-peer voice 1 pots destination-pattern 7001 port 2/0/0 ! dial-peer voice 2 vofr codec x bytes y destination-pattern 6001 session protocol frf11-trunk session target Sn 100 d called-number 7001 dtmf-relay vad	hostname mc3810B ! interface serial 0 ip address xxx.xxx.xxx 255.255.255.0 encapsulation frame-relay frame-relay traffic shaping frame-relay interface-dlci 100 class voice vofr data 4 ! map-class frame-relay voice frame-relay cir s frame-relay min-cir in x frame-relay bc u frame-relay voice bandwidth v frame-relay fragment y ! voice-port 1/5 connection trunk 7001 ! dial-peer voice 2 pots destination-pattern 6001 port 1/5 ! dial-peer voice 4 vofr codec x bytes y destination-pattern 7001 session protocol frf11-trunk session target Sn 100 d dtmf-relay vad

Tandem Configuration Examples

Figure 97 shows an example of a tandem configuration with two Cisco 3600 series routers as endpoints and a third Cisco 3600 series router as a tandem node.

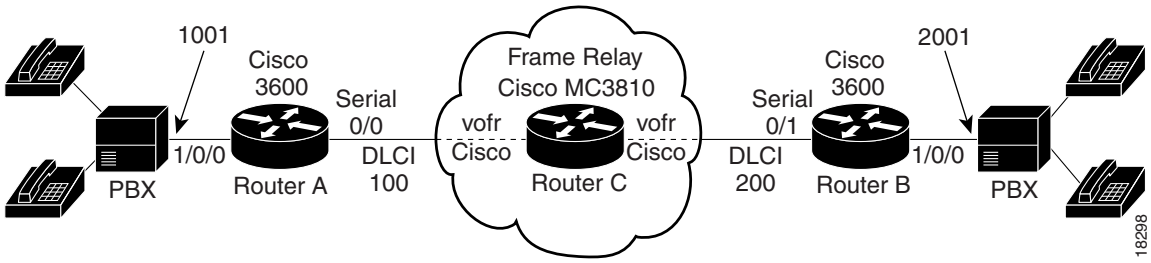
Figure 97 Tandem Configuration with Three Routers for Switched Calls



Router A Endpoint	Router C Tandem Node	Router B Endpoint
hostname 3600A ! interface serial 0/0 encapsulation frame-relay frame-relay traffic-shaping frame-relay interface-dlci 100 class voice vofr data 4 call-control 5 ! map-class frame-relay voice frame-relay cir a frame-relay min-cir t frame-relay bc b frame-relay voice bandwidth c frame-relay fragment d ! dial-peer voice 1 pots destination-pattern 1001 port 1/0/0 ! dial-peer voice 2 vofr destination-pattern 2... session target serial 0/0 100 ! voice-port 1/0/0	hostname3600C ! interface serial 0/0 encapsulation frame-relay frame-relay traffic-shaping frame-relay interface-dlci 100 class voice vofr data 4 call-control 5 ! interface serial 0/1 encapsulation frame-relay frame-relay traffic-shaping frame-relay interface-dlci 200 class voice vofr ! map-class frame-relay voice frame-relay cir a frame-relay min-cir t frame-relay bc b frame-relay voice bandwidth c frame-relay fragment d ! dial-peer voice 1 vofr destination-pattern 1... session target serial 0/0 100 ! dial-peer voice 2 vofr destination-pattern 2... session target serial 0/1 200	hostname3600B ! interface serial 0/0 encapsulation frame-relay frame-relay traffic-shaping frame-relay interface-dlci 100 class voice vofr data 4 call-control 5 ! map-class frame-relay voice frame-relay cir a frame-relay min-cir t frame-relay bc b frame-relay voice bandwidth c frame-relay fragment d ! dial-peer voice 1 pots destination-pattern 2001 port 1/0/0 ! dial-peer voice 2 vofr destination-pattern 1... session target serial 0/0 200 ! voice-port 1/0/0

Figure 98 shows an example of a tandem configuration with a Cisco MC3810 multiservice concentrator acting as a tandem node.

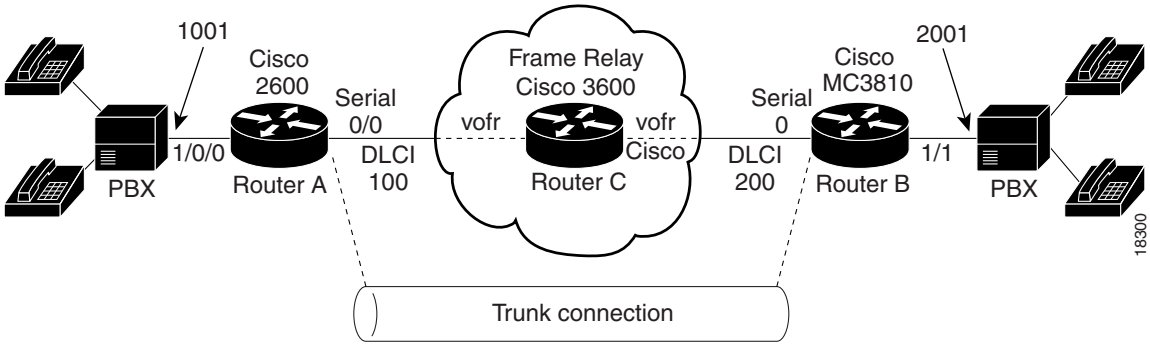
Figure 98 Tandem Configuration with a Cisco MC3810 Multiservice Concentrator Tandem Node for Switched Calls



Router A Endpoint	Router C Tandem Node	Router B Endpoint
interface serial 0/0	interface serial 0	interface serial 0/0
encapsulation frame-relay	encapsulation frame-relay	encapsulation frame-relay
frame-relay traffic-shaping	frame-relay traffic-shaping	frame-relay traffic-shaping
frame-relay interface-dlci 100	frame-relay interface-dlci 100	frame-relay interface-dlci 100
class voice	class voice	class voice
vofr data 4 call-control 5	vofr data 4 call-control 5	vofr data 4 call-control 5
!	!	!
map-class frame-relay voice	interface serial 1	map-class frame-relay voice
frame-relay cir a	encapsulation frame-relay	frame-relay cir a
frame-relay min-cir t	frame-relay traffic-shaping	frame-relay min-cir t
frame-relay bc b	frame-relay interface-dlci 200	frame-relay bc b
frame-relay voice bandwidth c	class voice	frame-relay voice bandwidth c
frame-relay fragment d	vofr data 4 call-control 5	frame-relay fragment d
!	!	!
dial-peer voice 1 pots	map-class frame-relay voice	dial-peer voice 1 pots
destination-pattern 1001	frame-relay cir a	destination-pattern 2001
port 1/0/0	frame-relay min-cir t	port 1/0/0
!	frame-relay bc b	!
!	frame-relay voice bandwidth c	!
dial-peer voice 2 vofr	frame-relay fragment d	dial-peer voice 2 vofr
destination-pattern 2...	!	destination-pattern 1...
session target serial 0/0 100	dial-peer voice 1 vofr	session target serial 0/0 200
!	destination-pattern 1...	!
voice-port 1/0/0	session target serial 0/0 100	voice-port 1/0/0
!	!	!
!	dial-peer voice 2 vofr	!
!	destination-pattern 2...	!!
!	session target serial 0/1 200	!

Figure 99 shows an example of a tandem configuration with a Cisco MC3810 multiservice concentrator acting as an endpoint node for Cisco trunk calls. When a Cisco MC3810 multiservice concentrator is on a VoFR network, the configuration for connections to and from the Cisco MC3810 multiservice concentrator is slightly different than for other routers that support VoFR. The **vofr cisco** command is required for those connections.

Figure 99 Tandem Configuration with a Cisco MC3810 Multiservice Concentrator Endpoint Node

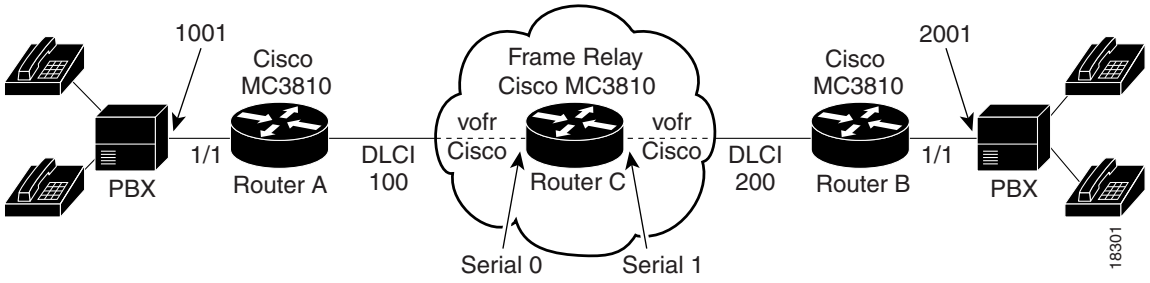


Router A Endpoint	Router C Tandem Node	Router B Endpoint
interface serial 0/0	interface serial 0/0	interface serial 0
encapsulation frame-relay	encapsulation frame-relay	encapsulation frame-relay
frame-relay traffic-shaping	frame-relay traffic-shaping	frame-relay traffic-shaping
frame-relay interface-dlci 100	frame-relay interface-dlci 100	frame-relay interface-dlci 200
class voice	class voice	class voice
vofr data 4 call-control 5	vofr data 4 call-control 5	vofr data 4 call-control 5
!	!	!
map-class frame-relay voice	interface serial 0/1	map-class frame-relay voice
frame-relay cir a	encapsulation frame-relay	frame-relay cir a
frame-relay min-cir t	frame-relay traffic-shaping	frame-relay min-cir t
frame-relay bc b	frame-relay interface-dlci 200	frame-relay bc b
frame-relay voice bandwidth c	class voice	frame-relay voice bandwidth c
frame-relay fragment d	vofr data 4 call-control 5	frame-relay fragment d
!	!	!
dial-peer voice 1 pots	map-class frame-relay voice	dial-peer voice 1 pots
destination-pattern 1001A	frame-relay cir a	destination-pattern 2001A
port 1/0/0	frame-relay min-cir t	port 1/1
!	frame-relay bc b	!
dial-peer voice 2 vofr	frame-relay voice bandwidth c	dial-peer voice 2 vofr
destination-pattern 2...	frame-relay fragment d	destination-pattern 1...
session target serial 0/0 100	!	session target serial 0 200
!	dial-peer voice 1 vofr	!
voice-port 1/0/0	destination-pattern 1...	voice-port 1/1
connection trunk 2001A answer-mode	session target serial 0/0 100	connection trunk 1001A
!	!	!
!	dial-peer voice 2 vofr	!
!	destination-pattern 2...	!
!	session target serial 0/1 200	!

Figure 100 shows an example of a tandem configuration with Cisco MC3810 multiservice concentrators as both endpoint and tandem nodes.

Note When a Cisco MC3810 multiservice concentrator running Cisco IOS software releases earlier than 12.1(2)T are used on a VoFR network, the configuration for connections to and from that Cisco MC3810 multiservice concentrator is slightly different from what is used for other routers that support VoFR. The **vofr cisco** command is required for these connections on the Cisco MC3810 multiservice concentrator.

Figure 100 Configuration with All Cisco MC3810 Multiservice Concentrators as Endpoint and Tandem Nodes



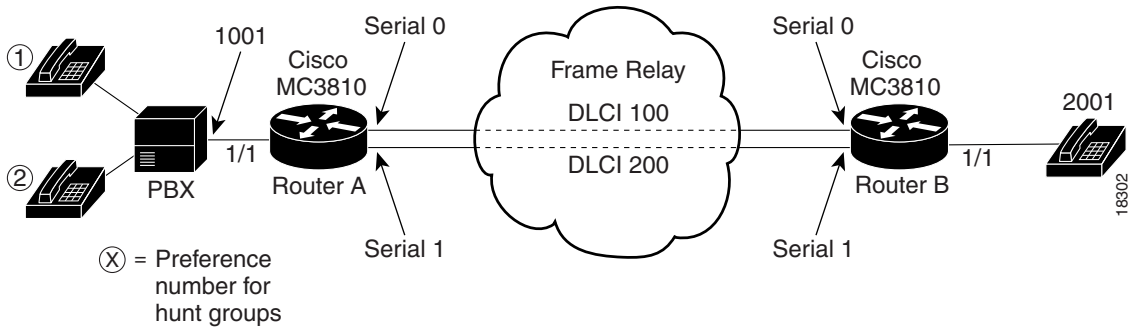
Router A Endpoint	Router C Tandem Node (Cisco IOS Releases Before 12.1(2)T)	Router B Endpoint
interface serial 0	interface serial 0	interface serial 0
encapsulation frame-relay	encapsulation frame-relay	encapsulation frame-relay
frame-relay traffic-shaping	frame-relay traffic-shaping	frame-relay traffic-shaping
frame-relay interface-dlci 100	frame-relay interface-dlci 100	frame-relay interface-dlci 200
class voice	class voice	class voice
vofr cisco	vofr cisco	vofr cisco
!	!	!
map-class frame-relay voice	interface serial 1	map-class frame-relay voice
frame-relay cir a	encapsulation frame-relay	frame-relay cir a
frame-relay bc b	frame-relay traffic-shaping	frame-relay bc b
frame-relay voice bandwidth c	frame-relay interface-dlci 200	frame-relay voice bandwidth c
frame-relay min-cir t	class voice	frame-relay fragment d
!	vofr cisco	frame-relay min-cir t
!	!	!
dial-peer voice 1 pots	map-class frame-relay voice	dial-peer voice 1 pots
destination-pattern 1001	frame-relay cir a	destination-pattern 2001
port 1/1	frame-relay min-cir t	port 1/1
!	frame-relay bc b	!
dial-peer voice 2 vofr	frame-relay voice bandwidth c	dial-peer voice 2 vofr
destination-pattern 2...	frame-relay fragment d	destination-pattern 1...
session target serial 0 100	!	session target serial 0 200
!	dial-peer voice 1 vofr	!
voice-port 1/1	destination-pattern 1...	voice-port 1/1
!	session target serial 0 100	!
!	!	!
!	dial-peer voice 2 vofr	!
!	destination-pattern 2...	!
!	session target serial 1 200	!

Cisco Trunk Call with Hunt Groups Example

Figure 101 shows an example of a Cisco trunk call with hunt groups configured. In this example, the two routers are in master-slave mode with a backup path. Router B is configured as a slave and Router A is configured as the master. The master makes periodic attempts to establish the trunk until the trunk is established.

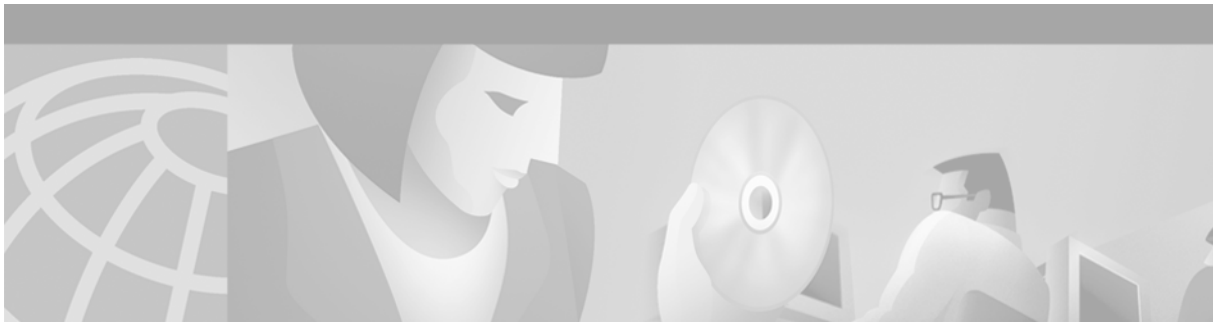
Two dial peers match the destination string configured in the voice port, but one dial peer has a higher preference, so the call setup is attempted through that dial peer. If the call setup fails, the master can continue attempting call setups using the next available dial peer. After all dial peers are exhausted, the master can continue following the list cyclically by starting again from the dial peer with the highest preference.

Figure 101 Cisco Trunk Call with Hunt Groups



Router A	Router B
interface serial 0	interface serial 0
encapsulation frame-relay	encapsulation frame-relay
frame-relay traffic-shaping	frame-relay traffic-shaping
frame-relay interface-dlci 100	frame-relay interface-dlci 100
class voice	class voice
vofr data 4 call-control 5	vofr data 4 call-control 5
!	!
interface serial 1	interface serial 1
encapsulation frame-relay	encapsulation frame-relay
frame-relay traffic-shaping	frame-relay traffic-shaping
frame-relay interface-dlci 200	frame-relay interface-dlci 200
class voice	class voice
vofr data 4 call-control 5	vofr data 4 call-control 5
!	!
map-class frame-relay voice	map-class frame-relay voice
frame-relay cir a	frame-relay cir a
frame-relay bc b	frame-relay bc b
frame-relay voice bandwidth c	frame-relay voice bandwidth c
frame-relay min-cir t	frame-relay min-cir t
!	!
dial-peer voice 1 pots	dial-peer voice 1 pots
destination-pattern 1001A	destination-pattern 2001A
port 1/1	port 1/1
!	!
dial-peer voice 100 vofr	dial-peer voice 100 vofr
destination-pattern 2...	destination-pattern 1...

Router A	Router B
session target serial0 100	session target serial0 100
preference 1	preference 1
!	!
dial-peer voice 200 vofr	dial-peer voice 200 vofr
destination-pattern 2...	destination-pattern 1...
session target serial1 200	session target serial1 200
preference 2	preference 2
!	!
voice-port 1/1	voice-port 1/1
connection trunk 2005A	description FXS port
description FXO port	connection trunk 1001A answer-mode
!	!
!	!



Configuring Voice over ATM

This chapter describes Voice over ATM (VoATM) and contains the following sections:

For a description of the VoATM commands, see the *Cisco IOS Voice, Video, and Fax Applications Command Reference*. For information about software configuration requirements for the Digital T1 Packet Voice trunk network modules on the Cisco 2600 and Cisco 3600, see the "Configuring Voice Ports" chapter. For more information about configuring ATM for data transmission, see the *Cisco IOS Wide-Area Networking Configuration Guide* and *Command Reference*.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

VoATM Overview

VoATM enables a router to carry voice traffic (for example, telephone calls and faxes) over an ATM network. An ATM network is a cell-switching and multiplexing technology designed to combine the benefits of circuit switching (constant transmission delay and guaranteed capacity) and packet switching (flexibility and efficiency for intermittent traffic).

All traffic to or from an ATM network is prefaced with a virtual path identifier (VPI) and virtual channel identifier (VCI). A VPI-VCI pair is considered a single virtual circuit. Each virtual circuit is a private connection to another node on the ATM network. Each virtual circuit is treated as a point-to-point mechanism to another router or host and is capable of supporting bidirectional traffic.

Each ATM node establishes a separate connection to every other node in the ATM network with which it must communicate. All such connections are established by means of a permanent virtual circuit (PVC) or a switched virtual circuit (SVC) with an ATM signaling mechanism. This signaling is based on the ATM Forum User-Network Interface (UNI) Specification V3.0.

Each virtual circuit is considered a complete and separate link to a destination node. Data can be encapsulated as needed across the connection, and the ATM network disregards the contents of the data. The only requirement is that data be sent to the ATM processor card of the router in a manner that follows the specific ATM adaptation layer (AAL) format.

An ATM connection transfers raw bits of information to a destination router or host. The ATM router takes the common part convergence sublayer (CPCS) frame, carves it up into 53-byte cells, and sends the cells to the destination router or host for reassembly. In AAL5 format, 48 bytes of each cell are used for the CPCS data and the remaining 5 bytes are used for cell routing. The 5-byte cell header contains the destination VPI-VCI pair, payload type, cell loss priority (CLP), and header error control (HEC) information.

AAL Technology

AAL defines the conversion of user information into the ATM cells. AAL protocols perform a convergence function; that is, they take whatever traffic is to be sent across the ATM network, establish the appropriate connections, and then package the traffic received from the higher layers into the 48-byte information payload that is passed down to the ATM layer for transmission. At the receiving level, the AAL layer must receive the information payloads passed up from the ATM layer and put the payloads into the form expected by the higher layer.

The AAL layers provide a service to the higher layers that corresponds to the four classes of traffic. AAL1 and AAL2 handle isochronous traffic, such as voice and video, but are not relevant to the router. AAL3/4 and AAL5 support data communications by segmenting and reassembling packets.

AAL2 is a bandwidth-efficient, standards-based trunking method for transporting compressed voice, voice-band data, circuit-mode data, and frame-mode data. VoATM with AAL2 trunking provides the following functionality:

- Increased quality of service (QoS) capabilities
- Robust architecture
- Signalling transparency
- CAS and CCS support

AAL5 is designed to support only message-mode, nonassured operation. AAL5 packets contain 48 bytes of data and a 5-byte header.

Variable Bit Rate Real-Time Options for Traffic Shaping

Variable bit rate (VBR) is a QoS class defined by the ATM Forum for ATM networks. VBR is subdivided into a real-time (RT) class and nonreal time (NRT) class. RT VBR is used for connections in which there is a fixed timing relationship between samples, as in the case of traffic shaping. NRT VBR is used for connections in which there is no fixed timing relationship between samples, but which still need a guaranteed QoS.

Traffic shaping prevents a carrier from discarding incoming calls from a Cisco router. Traffic shaping is performed by configuring the peak, average, and burst options for voice traffic. Burst is required if the PVC is carrying bursty traffic. Peak, average, and burst are required so the PVC can effectively handle the bandwidth for the number of voice calls.

Cisco Trunk Calls on Cisco MC3810 Multiservice Concentrators

Cisco trunk (private-line) calls are basically dynamic switched calls of indefinite duration that use a fixed destination telephone number and include optional transparent end-to-end signaling. The telephone number of the destination endpoint is permanently configured into the router so that it always selects a fixed destination. After the call is established, either at boot-up or when configured, the call stays up until one of the voice ports or network ports is shut down or a network disruption occurs.

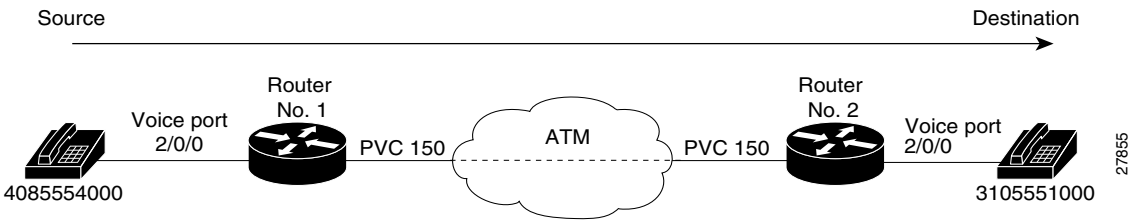
The Cisco trunk call functionality provides the following benefits:

- True permanent, private-line connections.
- Comprehensive busyout support for trunk connections. For more information, see to the "Configuring Trunk Connections and Conditioning Features" chapter.
- Transparent CAS protocol transport to enable the trunk to carry arbitrary ABCD signaling protocols.
- Conversion from North American signaling protocols to CEPT (Conférence Européenne des Postes et des Télécommunications) signaling protocols used for European voice networks.
- Remote analog-to-digital channel-bank operation for converting from digital voice multiplexer (DVM) to ATM voice multiplexer (AVM) configurations on the Cisco MC3810 multiservice concentrator.

VoATM Dial Peers

Establishing two-way communications using VoATM requires a specific voice connection between two defined endpoints. As shown in Figure 102, the plain old telephone service (POTS) dial peer establishes the source (the originating telephone number and voice port) of the call, and the VoATM dial peer establishes the destination by associating the destination phone number with a specific ATM virtual circuit.

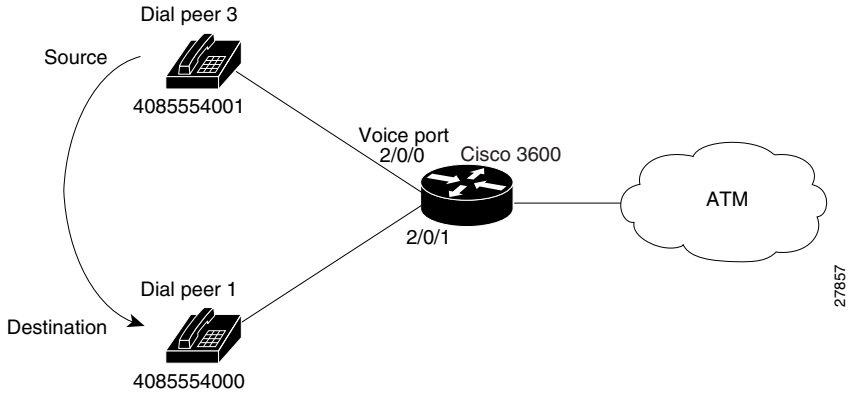
Figure 102 Calls from the Perspective of Router 1



In Figure 102 the destination string, 14085554000, coming from the source, maps to U.S. phone number 555-4000, with the digit "1" plus the area code "408" preceding the number. When configuring the destination pattern, set the dial string to match the local dial conventions.

When both POTS dial peers are connected to the same router and share the same destination IP address, the VoATM dial peer does not need to be configured (see Figure 103).

Figure 103 Communication Between Dial Peers Sharing the Same Router



When configuring VoATM dial peers, an understanding of the relationship between the destination pattern and the session target is critical. The destination pattern represents the pattern for the device at the voice connection endpoint, such as a telephone or a PBX. The session target represents the serial port on the peer router at the other end of the ATM connection. Figure 104 and Figure 105 show the relationship between the destination pattern and the session target, as seen from the perspective of both routers in a VoATM configuration.

Figure 104 Relationship Between the Destination Pattern and Session Target from the Perspective of Router 1

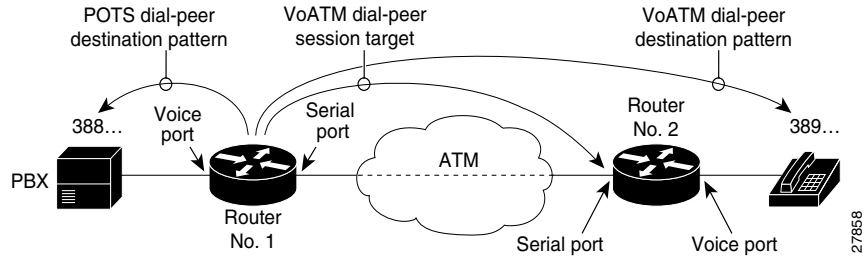
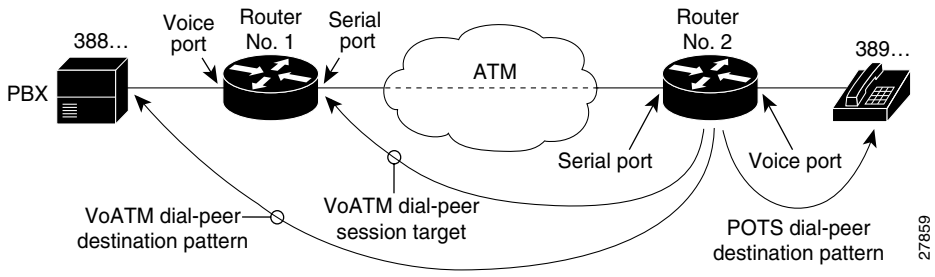


Figure 105 Relationship Between the Destination Pattern and Session Target from the Perspective of Router 2



For more information regarding dial peers, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

VoATM Restrictions

The following are restrictions regarding V0ATM:

- VoATM is supported only on the Cisco MC3810 multiservice concentrators ATM port 0 (compressed VoATM). When VoATM is enabled, the channel group, time-division multiplexing (TDM) group, and channel associated signaling (CAS) functionality are not available on the multiflex trunk (MFT) because ATM uses all T1/E1 time slots.
- VoATM on the Cisco 3600 series router requires the installation of one of the following modules:
  - Multiport T1/E1 ATM network module with inverse multiplexing over ATM (IMA). The multiport T1/E1 ATM network module with IMA supports up to eight T1/E1 lines. For more information, see the Cisco IOS Release 12.0(5)T online document *Configuring Multiport T1/E1 ATM Network Modules with Inverse Multiplexing over ATM on Cisco 2600 and 3600 Series Routers.*
  - OC3 ATM network module. The OC3 ATM network module supports one OC3 line. For more information about the Digital T1 packet voice trunk network modules, see the Cisco IOS Release 12.0(3)T online document *ATM OC-3 Network Module for the Cisco 3600 Series Routers.*
- The following AAL2 capabilities are not supported:
  - Data services over AAL2 (Nx64K circuit mode and N>=1)
  - Fax/modem relay (fax demodulation and remodulation)
  - Idle code detection or idle channel suppression
  - Cisco-switched AAL2 trunking
- Only AAL5 is supported on the Cisco 3600 series routers. AAL2 is not supported.
- VoATM SVCs are not supported on the Cisco 3600 series routers.

VoATM Prerequisite Tasks

Before configuring VoATM, perform the following tasks:

- Install the required network modules into the Cisco 3600 series router. For more information, see the "VoATM Restrictions" section on page 461.
- Establish a working ATM network. For more information, refer to the *Cisco IOS Wide-Area Networking Configuration Guide.*
- Configure Local Management Interface (LMI) support if the carrier is using LMI because ATM defaults to Integrated Local Management Interface (ILMI).
- Configure the clock source for the Cisco MC3810 multiservice concentrator interfaces. For more information, see the "Configuring Synchronous Clocking on the Cisco MC3810 Multiservice Concentrators" appendix.

- Complete your company dial plan and establish a working telephony network based on the plan and:
  - Integrate the dial plan and telephony network into the existing ATM network topology. Make routing and dialing transparent to the user; for example, avoid secondary dial tones from secondary switches where possible.
  - Contact the PBX vendor for instructions about how to reconfigure the appropriate PBX interfaces.
- Ensure that the voice ports and dial peers are configured. For more information, see the "Configuring Voice Ports" and "Configuring Voice Dial Peers, Dial Plans, and Digit Manipulation" chapters.

VoATM Configuration Task List

To configure VoATM, perform the following tasks:

- Configuring ATM Interfaces for Voice Traffic Using AAL5, page 462
- Configuring AAL2 Encapsulation for VoATM, page 465
- Configuring VoATM Dial Peers, page 469
- Configuring Dial-Peer Hunting, page 474
- Configuring Cisco Trunk Permanent Calls, page 475
- Configuring Cisco Trunk Permanent Calls, page 475

Configuring ATM Interfaces for Voice Traffic Using AAL5

ATM interfaces must be configured for voice traffic using AAL5 and the VoATM configuration must be performed on both sides of the voice connection. The only commands in ATM virtual circuit configuration mode that are used for ATM voice PVCs are **encapsulation aal5mux voice**, **vbr-rt**, and **ilmi**. For more information on the encapsulation command, see the *Cisco IOS Wide-Area Networking Command Reference*.

To calculate the *minimum* peak, average, and burst values for the number of voice calls, perform the following calculations:

- Peak value: (2 x the maximum number of calls) x 16 Kb
- Average value: (1 x the maximum number of calls) x 16 Kb
  - The average value correlates to the carrier sustainable cell rate (SCR).
- Burst value: 4 x the maximum number of calls
  - The burst value is the burst size in cells.

To configure ATM interfaces to support voice traffic, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface ATM slot/number`	Enters ATM interface configuration mode.
Step 2	`Router(config-if)# pvc [name] vpi/vci [ilmi | qsaal | smds]`	Creates an ATM PVC for voice traffic and enters virtual circuit configuration mode. The keywords and arguments are as follows: • *name*—(Optional) Supports up to 16 characters. • *vpi/*—Valid range is from 0 to 255. • *vci*—Valid range is from 0 to 1 less than the maximum value set for the interface by the atm **vc-per-vp** command. Lower values 0 to 31 are reserved fro specific traffic (for example, F4 OAM, SVC signaling, ILMI, etc.) and should not be used. VCI is a 16-bit field in the header of the ATM cell. The value is unique only on a single link. not throughout the ATM network, because it has local significance only. **Note** The vpi and vci arguments cannot both be set to 0. • **ilmi**—(Optional) Sets up communication with the ILMI. The *vpi* and *vci* values are 0 and 16, respectively. • **qsaal**—(Optional) Signalling-type PVC used for setting up or tearing down SVCs. The associated *vpi* and *vci* values are 0 and 5, respectively. The default is that the PVC is not defined. When the PVC is defined, the global default of the **encapsulation** command applies (aal-encap = aal5snap).
Step 3	`Router(config-if-atm-pvc)# encapsulation aal5mux voice`	Sets the encapsulation of the PVC to support AAL5 voice.

	Command	Purpose
Step 4	`Router(config-if-atm-pvc)# vbr-rt peak-rate average-rate [burst]`	Configures the peak and average rates and burst cell size to perform traffic shaping between voice and data PVCs for real-time voice networks. The arguments are as follows: • *peak rate*—Sets to the line rate if it does not exceed the carrier allowable rate (for example, 1536 kbps for T1-ATM). • *average rate*—Calculates according to the maximum number of PVC calls times the bandwidth per call. The following formulas calculate the average rate in kbps: – G.711 with 40- or 80-byte sample size: maximum calls x 85 – G.726 with 40- or 80-byte sample size: maximum calls x 43 – G.729 with 30-byte sample size: maximum calls x 15 – G.729 with 20-byte sample size: maximum calls x 22 – G.729 with 10-byte sample size: maximum calls x 43 If VAD is enabled, the bandwidth usage is reduced by as much as 12 percent with the maximum number of calls in progress. With fewer calls in progress, bandwidth is less. • *burst* (Optional)—Sets the burst size as large as possible and never less than the minimum burst size. Guidelines are as follows: – Minimum: number of voice calls x 4. – Maximum: maximum allowed by the carrier.
Step 5	`Router(config-if-atm-pvc)# exit`	Exits ATM virtual circuit configuration mode.
Step 6	`Router(config-if)# pvc [name] vpi/vci`	Creates an ATM PVC for data traffic and enters virtual circuit configuration mode.
Step 7	`Router(config-if-atm-pvc)# encapsulation aal5snap`	Sets the encapsulation of the PVC to support ATM data traffic. In ATM PVC configuration mode, configure the **ubr**, **ubr+** or the **vbr-nrt** traffic shaping commands for the data PVC as appropriate. **Note** Calculate the overhead as voice rate x 1.13. See the *Cisco IOS Wide-Area Network Configuration Guide* for more information.
Step 8	`Router(config-if-atm-pvc)# exit`	Exits ATM virtual circuit configuration mode. Repeat Steps 6 and 7 for each data PVC configured.

Verifying the ATM PVC Configuration

Verify the ATM PVC configuration by using the **show atm vc** command. To verify connectivity, do not use the **ping** command over a voice PVC because the command applies to data only. Use the **ping** command over the data PVC to verify that the data and voice PVCs are set to the same destination.

Configuring AAL2 Encapsulation for VoATM

AAL2 encapsulation for VoATM must be configured and the VoATM configuration must be performed on the Cisco MC3810 multiservice concentrators at both ends of the ATM link. AAL2 is not supported on the Cisco 3600 series routers.

Note If any DS0 groups (CAS groups), channel groups, or clear channels are configured on T1/E1 controller 0, remove them before configuring VoATM. Because ATM uses all the DS0 timeslots on the controller, the ATM configuration cannot take place if any DS0s on controller 0 are used by other applications.

To configure AAL2 encapsulation for VoATM, perform the following tasks:

- Configuring T1/E1 Trunks, page 465
- Configuring Call Admission Control, page 467
- Configuring Subcell Multiplexing, page 468

Configuring T1/E1 Trunks

To configure the T1/E1 trunk, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {t1 | e1} 0`	Selects the T1 or E1 controller 0. **Note** On the Cisco MC3810 multiservice concentrator, ATM is supported only on controller 0.
Step 2	`Router(config-controller)# mode atm`	Specifies controller support for ATM encapsulation and creates ATM interface 0. When the controller is set to ATM mode, the following takes place: • Controller framing is automatically set to Extended SuperFrame (ESF) on T1 and to CRC4 on E1. • The linecode is automatically set to B8ZS on T1 and to HDB3 on E1.
Step 3	`Router(config-controller)# no shutdown`	Ensures that the controller is activated.

	Command	Purpose
Step 4	`Router(config)# interface atm0 [subinterface-number [multipoint | point-to-point]]`	Enters interface configuration mode to configure ATM interface 0 or an ATM subinterface. The keywords and arguments are as follows: • *subinterface-number*—Configures the subinterface. Valid range is from 1 to 4294967293. • **multipoint** (Optional)—Assumes that is a fully meshed network. This is the default setting. • **point-to-point** (Optional)—Specifies the VoATM connection over point-to-point network.
Step 5	`Router(config-if)# pvc [name] {vpi/vci | vci}`	Creates an ATM PVC for voice traffic and enters ATM virtual circuit configuration mode. **Note** The **ilmi** and **qsaal** options are not supported for AAL2.
Step 6	`Router(config-if-atm-pvc)# encapsulation aal2`	Sets the PVC encapsulation to support AAL2 voice traffic. This automatically creates channel identifiers (CIDs) 1 through 255.
	or	
	`Router(config-if-atm-pvc)# ip address ip-address mask`	Assigns the IP address and subnet mask to the interface on the Cisco MC3810 multiservice concentrator.
	and	
	`Router(config-if-atm-pvc)# encapsulation aal5mux`	Sets the encapsulation of the PVC to support voice traffic on the Cisco MC3810 multiservice concentrator.
Step 7	`Router(config-if-atm-pvc)# vbr-rt peak-rate average-rate [burst]`	Configures the VBR for real-time voice traffic.

	Command	Purpose
Step 8	`Router(config-if-atm-pvc)# oam-pvc [manage] [frequency]`	(Optional) Configures transmission of end-to-end F5 operation, administration, and maintenance (OAM) loopback cells on a PVC; specifies the number of seconds between loopback cells; and enables OAM management of the connection. The keyword and argument are as follows: • **manage**—(Optional) Enables OAM management. • *frequency* (Optional)—Valid range is 0 to 600. The default is 10. **Note** The **oam-pvc** command does not apply to AAL2.
Step 9	`Router(config-if-atm-pvc)# oam retry up-count down-count retry-frequency`	(Optional) Specifies OAM management parameters for verifying connectivity of a PVC connection. This command is supported only if OAM management is enabled. The arguments are as follows: • *up-count*—Number of OAM loopback cell responses received to change the PVC connection to up. The range is from 1 to 600; the default is 3. • *down-count*—Number of OAM loopback cell responses not received to change the PVC connection to down. The range is from 1 to 600; the default is 5. • *retry-frequency*—Number of seconds between loopback cells sent to verify the down state of a PVC. The range is from 1 to 1000; the default is 1. **Note** Enter the **oam retry** command only once with all of the arguments in the order shown. The first number always specifies *up-count*; the second, *down-count*; and the third, *retry-frequency.* **Note** The **oam retry** command does not apply to AAL2.

Configuring Call Admission Control

Configuring the call admission control (CAC) is optional for the Cisco MC3810 multiservice concentrator because the MC3810 multiservice concentrator can be configured as master or slave. By default, a Cisco MC3810 multiservice concentrator is a CAC slave.

Typically the ATM trunk is configured with the CAC master at one end (performing CAC during fax/modem up speed) and slave at the opposite end. When the Cisco MC3810 multiservice concentrator is configured as a slave, it sends a request for CAC to the CAC master.

To configure a Cisco MC3810 multiservice concentrator as a CAC master, usethe following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice service voatm`	Enters voice-service configuration mode.
Step 2	`Router(config-voice-service)# session protocol aal2`	Enters voice-service-session configuration mode and specifies AAL2 trunking.

	Command	Purpose
Step 3	`Router(config-voice-service-session)# cac master`	Configures the Cisco MC3810 multiservice concentrator as a CAC master. Default is that the concentrator acts as a CAC slave.
Step 4	`Router(config-voice-service-session)# exit`	Exits voice-service session configuration mode. To return to global configuration mode, enter the **exit** command again.

To return a Cisco MC3810 multiservice concentrator to its default operation as a CAC slave, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice service voatm`	Enters voice-service configuration mode.
Step 2	`Router(config-voice-service)# session protocol aal2`	Enters voice-service-session configuration mode and specifies AAL2 trunking.
Step 3	`Router(config-voice-service-session)# no cac master`	Configures this Cisco MC3810 multiservice concentrator as a CAC slave.
Step 4	`Router(config-voice-service-session)# exit`	Exits voice-service session configuration mode. To return to global configuration mode, enter the **exit** command again.

Configuring Subcell Multiplexing

This section describes the configuration tasks necessary to enable AAL2 common part sublayer (CPS) subcell multiplexing when the Cisco MC3810 multiservice concentrator interoperates with a voice interface service module (VISM) in an MGX switch. The commands and procedures in this section are specific to the Cisco MC3810 multiservice concentrator.

To configure the Cisco MC3810 multiservice concentrator to perform subcell multiplexing, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice service voatm`	Enters voice-service configuration mode.
Step 2	`Router(config-voice-service)# session protocol aal2`	Enters voice-service-session configuration mode and specifies AAL2 trunking.
Step 3	`Router(config-voice-service-session)# subcell-mux`	Enables subcell multiplexing. By default, subcell multiplexing is not enabled.
Step 4	`Router(config-voice-service-session)# exit`	Exits voice-service session configuration mode. To return to global configuration mode, enter the **exit** command again.

Configuring VoATM Dial Peers

Configuring dial peers to support VoATM should be performed in a back-to-back configuration before separating them across the ATM network. The back-to-back configuration enables the testing of a voice connection. If a voice connection cannot be made after both peers are placed in the network, then you have a network problem. For information about configuring POTS dial peers, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter.

To configure VoATM dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `voatm`	Defines a VoATM dial peer and enters dial-peer configuration mode. The *number* argument identifies the dial peer. Do not duplicate a specific number.
Step 2	`Router(config-dial-peer)# destination-pattern` *string*	Configures the destination pattern. The special characters are as follows: The string values are as follows: • Asterisk (*) or pound sign (#) can be used, but not as leading characters (for example, *650 would be unacceptable). • Period (.) can be entered as a wildcard digit. Network dial peers typically use wildcards to represent a range of destination telephone numbers (for example, 1408555... for all numbers in area code 408 with a 555 prefix). • Comma (,) can be used only in prefixes and inserts a one-second pause. • Timer (T) can be used to configure variable-length dial plans.
Step 3	`Router(config-dial-peer)# session target ATM` *x/y* `pvc` {*name* | *vpi/vci* | *vci*}	Configures the ATM session target. On the Cisco 3600, if a *vpi*/*vci* combination is specified, the valid values depend on the network module installed, as follows: • For multiport T1/E1 ATM with IMA, the valid ranges are: – *vpi* is from 0 to 15 – *vci* is from 1 to 255 • For OC3 ATM, the valid ranges are: – *vpi* is from 0 to 15 – *vci* is from 1 to 1023
Step 4	`Router(config-dial-peer)# preference` *value*	(Optional) Configures a preference. The *value* argument has a valid range is from 0 to 10 (the lower the number, the higher the preference).

	Command	Purpose
Step 5	`Router(config-dial-peer)# codec type [bytes payload_size]`	Specifies the rate of speech and payload size. The default codec is **g729r8**. The keyword and arguments are as follows: • *type*—Assigns codec values to the voice port for regular switched voice calls. • **bytes**—(Optional) Specifies the payload size. Each codec type defaults to a different payload size if one is not specified. • *payload_size*—(Optional) Specifies the payload size by entering the bytes. Each codec type defaults to a different payload size if a value is not specified. **Note** To obtain a list of the default payload sizes, enter the **codec** command and the **bytes** option followed by a question mark (**?**).
Step 6	`Router(config-dial-peer)# dtmf-relay`	(Optional) Specifies support for dual tone multifrequency (DTMF) relay. If the codec type is a low bit-rate codec such as g729 or g723, the end-to-end transport of DTMF tones is improved. DTMF tones do not always propagate reliably with low bit-rate codecs. DTMF relay is disabled by default.
Step 7	`Router(config-dial-peer)# signal-type {cas | cept | ext-signal | transparent}`	(Optional) Defines the ABCD signaling packets that are generated by the voice port and sent to the data network. The signal type must be configured to the same setting at both ends of the permanent voice call. The keywords are as follows: • **cas**—Support for CAS. • **cept**—Support for the European CEPT standard (related to Mercury Exchange Limited (MEL) CAS). • **ext-signal**—Indicates that ABCD signaling packets should not be sent for configurations in which the line signaling information is carried externally to the voice port. • **transparent**—(for digital T1/E1 interfaces) Reads the ABCD signaling bits directly from the T1/E1 interface without interpretation and transparently passes them to the data network. Also known as transparent FRF.11 signaling.
Step 8	`Router(config-dial-peer)# no vad`	(Optional) Disables voice activity detection (VAD). This command is enabled by default.
Step 9	`Router(config-dial-peer)# sequence-numbers`	(Optional) Enables the voice sequence number if required. This command is disabled by default.

	Command	Purpose
Step 10	`Router(config-dial-peer)# preference value`	(Optional) Configures a preference for the VoATM dial peer. The *value* argument has valid ranges from 0 to 10 (the lower the number, the higher the preference in hunt groups).
Step 11	`Router(config-dial-peer)# session protocol cisco-switched`	(Optional) Configures the session protocol to support Cisco-trunk permanent trunk calls. The **cisco-switched** keyword is the default setting and is not required. **Note** Use the **no session protocol cisco-switched** command if the dial peer does not support Cisco trunk calls.
Step 12	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode. Repeat the steps to configure each dial peer.

Configuring VoATM Dial Peers to Support AAL2

To configure the voice network dial peers to support AAL2 on a Cisco MC3810 multiservice concentrator, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number voatm`	Defines the dial peer and enters dial-peer configuration mode.
Step 2	`Router(config-dial-peer)# destination-pattern string`	Configures the destination pattern.
Step 3	`Router(config-dial-peer)# session protocol aal2-trunk`	Configures the session protocol to support AAL2-trunk permanent trunk calls.
Step 4	`Router(config-dial-peer)# session target atm 0 pvc {name | vpi/vci | vci}`	Configures the ATM session target for the dial peer. Be sure to specify **atm 0** as the interface for the PVC.

	Command	Purpose
Step 5	`Router(config-dial-peer)# codec aal2 profile {itut | custom} profile-number codec`	Specifies a codec profile for the DSP. Use this command instead of the **codec (dial-peer)** command for AAL2 trunk applications. The keywords and arguments are as follows: • **itut—**Specifies the *profile-number* as an ITU-T type: – 1: G.711ulaw – 2: G.711ulaw with silence insertion descriptor (SID) – 7: G.711ulaw and G.729ar8 • **custom—**Specifies the *profile-number* as a custom type: – 100: G.711ulaw and G.726r32 – 110: G.711ulaw, G.726r32, and G.729ar8 • *profile-number*—The available *profile-number* selections depend on the profile type. • *codec*—Enter one codec for the domain specific part (DSP). The possible *codec* entries depend on the *profile-number*. The valid entries are as follows: – For ITU 1: g711ulaw – For ITU 2: g711ulaw – For ITU 7: g711ulaw or g729ar8 – For custom 100: g711ulaw or g726r32 – For custom 110: g711ulaw or g726r32 or g729ar8 See the *Cisco IOS Voice, Video, and Fax Command Reference* for the codec options available for each AAL2 profile.
Step 6	`Router(config-dial-peer)# dtmf-relay`	(Optional) Specifies support for DTMF relay to improve end-to-end transport of DTMF tones if the codec type is a low bit-rate codec such as g729 or g723. DTMF tones do not always propagate reliably with low bit-rate codecs. DTMF relay is disabled by default.

	Command	Purpose
Step 7	`Router(config-dial-peer)# signal-type {ext-signal | transparent}`	(Optional) Defines the type of ABCD signaling packets that are generated by the voice port and sent over the ATM network. The signal type must be configured to the same setting at both ends of the PVC. The keywords are as follows: • **ext-signal**—Identifies common-channel signaling (CCS). ABCD signaling packets are not sent. • **transparent**—Identifies nonswitched trunks using channel associated signaling (CAS). ABCD signaling bits are passed transparently to the ATM network.
Step 8	`Router(config-dial-peer)# no vad`	(Optional) Disables VAD on the dial peer. VAD is enabled by default.
Step 9	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode. Repeat the steps to configure each dial peer.

Configuring VoATM Dial Peers for Cisco Trunk Calls

If Cisco trunk calls are transmitted over ATM, the dial peers must be configured to specifically support the calls. Cisco trunk calls are permanent calls.

Note A voice class to configure trunk conditioning values for the idle and out-of-service (OOS) states can be configured with the voice class assigned to the VoATM dial peer. For more information, see the "Configuring Trunk Management Features" chapter.

To configure a VoATM dial peer to support Cisco trunk calls, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number voatm`	Defines a VoATM dial peer and enters dial-peer configuration mode. **Note** The VoATM dial peers must already be configured.
Step 2	`Router(config-dial-peer)# session protocol cisco-switched`	Configures the session protocol to support Cisco trunk calls.

Configuring Dial-Peer Hunting

To configure dial-peer hunting, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer hunt` *hunt-order-number*	Specifies the hunt selection order for dial peers. The *hunt-order-number* has valid ranges from 0 to 7 as follows: • 0: Longest match in phone number, explicit preference, random selection. This is the default hunt order number. • 1: Longest match in phone number, explicit preference, least recent use. • 2: Explicit preference, longest match in phone number, random selection. • 3: Explicit preference, longest match in phone number, least recent use. • 4: Least recent use, longest match in phone number, explicit preference. • 5: Least recent use, explicit preference, longest match in phone number. • 6: Random selection. • 7: Least recent use. The default is the longest match in a phone number, explicit preference, and random selection (hunt order number 0).
Step 2	`Router(config)# dial-peer terminator` *character*	(Optional) Designates a special character for variable length dialed numbers. The character argument has valid numbers and characters that are as follows: • Pound sign (#) • Asterisk (*) • Numbers from zero to nine • Letters from a to d The default is #.

To disable dial-peer hunting, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* {**pots** | **voatm**}	Enters dial-peer configuration mode for the specified dial peer.
Step 2	`Router(config-dial-peer)# huntstop`	Disables dial-peer hunting on the dial peer. No further hunting is enabled if a call fails on the specified dial peer. **Note** To reenable dial-peer hunting on a dial peer, enter the **no huntstop** command.

Configuring Cisco Trunk Permanent Calls

The Cisco trunk call functionality provides true permanent, private-line connections; comprehensive busyout support for trunk connection; and transparent CAS protocol transport to allow the trunk to carry arbitrary ABCD signaling protocols. Conversion from North American signaling protocols to CEPT (Conférence Européenne des Postes et des Télécommunications) signaling protocols used for European voice networks and remote analog to digital channel-bank operation for converting from DVM to AVM configurations is also provided.

To configure Cisco-trunk permanent calls, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice-port`	Enters voice-port configuration mode. **Note** The **voice-port** command is hardware specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference* for more information.
Step 2	`Router(config-voiceport)# connection trunk` *destination-string* [**answer-mode**]	Configures the trunk connection, specifying the telephone number in the *destination-string* argument. The **answer-mode** keyword specifies that the voice port should operate in slave mode. The default is master mode.
Step 3	`Router(config-voiceport)# shutdown`	Shuts down the voice port.
Step 4	`Router(config-voiceport)# no shutdown`	Reactivates the voice port to enable the trunk connection to take effect.

Note When the **connection trunk** or **no connection trunk** command is entered, the voice port must be toggled by entering **shutdown**, and then **no shutdown** before the changes take effect.

Verifying the Voice Connection

To verify that the voice connection is working, perform the following steps:

Step 1 Pick up the telephone handset and verify that a dial tone is present.

Step 2 Make a call from the local telephone to a configured dial peer and verify that the call attempt is successful.

Step 3 Use the **show dial-peer voice** command to verify that the configured data is correct.

Step 4 Use the **show voice port** command to show the status of the voice ports.

Step 5 Use the **show voice call** command to show the call status for all voice ports.

Step 6 Use the **show voice dsp** command to show the current status of all DSP voice channels.

Troubleshooting Tips

To resolve suspected problems, perform the following tasks:

Step 1 Use the **show dial-peer voice** command on the local and remote concentrators to verify that the data is configured correctly on both.

Step 2 Use the **show interface** command to verify that the ATM interface is up.

Step 3 Ensure that the voice port, serial port, and controller T1 0 is set to **no shutdown**.

Note ATM defaults to Interim Local Management Interface (ILMI). If the carrier is using LMI, be sure to configure LMI support on the router.

Verifying the ATM Interface Configuration

To verify the ATM interface configuration, perform the following tasks:

- Enter the privileged EXEC **show atm vc** command to view the SVCs and PVCs set. The following is a sample output:

```
Router# show atm vc

VCD /                                          Peak  Avg/Min Burst
Interface   Name       VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
0           1            0     5  PVC    SAAL     UBR       0                UP
0           2            0    16  PVC    ILMI     UBR       0                UP
0           379          0    60  SVC    SNAP     UBR       0                UP
0           986          0    84  SVC    SNAP     UBR       0                UP
0           14           0   133  SVC    VOICE    VBR      64     16   10    UP
0           15           0   134  SVC    VOICE    VBR      64     16   10    UP
0           16           0   135  SVC    VOICE    VBR      64     16   10    UP
0           17           0   136  SVC    VOICE    VBR      64     16   10    UP
0           18           0   137  SVC    VOICE    VBR      64     16   10    UP
0           19           0   138  SVC    VOICE    VBR      64     16   10    UP
```

```
0          20          0   139   SVC    VOICE    VBR      64     16   10    UP
0          21          0   140   SVC    VOICE    VBR      64     16   10    UP
0          22          0   141   SVC    VOICE    VBR      64     16   10    UP
0          23          0   142   SVC    VOICE    VBR      64     16   10    UP
0          24          0   143   SVC    VOICE    VBR      64     16   10    UP
0          25          0   144   SVC    VOICE    VBR      64     16   10    UP
0          26          0   145   SVC    VOICE    VBR      64     16   10    UP
0          27          0   146   SVC    VOICE    VBR      64     16   10    UP
0          28          0   147   SVC    VOICE    VBR      64     16   10    UP
```

- Enter the **show atm svc** command with or without the VPI/VCI specified. The following is a sample output for the specific SVC:

```
Router# show atm svc 0/134

ATM0: VCD: 5, VPI: 0, VCI: 134
VBR, PeakRate: 64000
AAL5, etype: 0x0, Flags 0x440, VCmode: 0xE000
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not Managed
ILMI VC state: Not Managed
InARP DISABLED
InPkts: 4, OutPkts: 4, InBytes: 432, OutBytes: 432
InPRoc: 4, OutPRoc: 4, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI:0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI:0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
OAM cell drops: 0
Status: UP
TTL: 3
interface = ATM0, call locally initiated, call reference = 5558610
vcnum = 5, vpi = 0, vci = 134, state = Active(U10), point-to-point call
Retry count: Current = 0
timer currently inactive, timer value = 00:00:00
Remote Atm Nsap address:47.00918100000000400B0A2501.0060837B4743.00, VCowner:Static
Map
```

- Enter the **show atm pvc** command with the VPI/VCI specified to view the PVCs that are set up for ILMI management and Q.SAAL signaling. The following is a sample output:

```
Router# show atm pvc 0/5

ATM0: VCD: 2, VPI: 0, VCI: 5, Connection Name: SAAL
UBR, PeakRate: 56
AAL5-SAAL, etype:0x4, Flags: 0x26, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s), OAM retry frequenc
y: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not Managed
ILMI VC state: Not Managed
InARP DISABLED
InPkts: 2044, OutPkts: 2064, InBytes: 20412, OutBytes: 20580
InPRoc: 2044, OutPRoc: 2064, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
```

```
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Compress: Disabled
Status: INACTIVE, State: NOT_IN_SERVICE
!
Router# show atm pvc 0/16

ATM0: VCD: 1, VPI: 0, VCI: 16, Connection Name: ILMI
UBR, PeakRate: 56
AAL5-ILMI, etype:0x0, Flags: 0x27, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s), OAM retry frequenc
y: 1 second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not Managed
ILMI VC state: Not Managed
InARP DISABLED
InPkts: 398, OutPkts: 421, InBytes: 30493, OutBytes: 27227
InPRoc: 398, OutPRoc: 421, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Compress: Disabled
Status: INACTIVE, State: NOT_IN_SERVICE
```

- Enter the **show atm interface** command in privileged EXEC mode and specify ATM 0 to display the ATM interface. The following is a sample output:

```
Router# show interface atm 0

ATM0 is up, line protocol is up
  Hardware is PQUICC Atom1
  Internet address is 9.1.1.6/8
  MTU 1500 bytes, sub MTU 1500, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 22/255, rxload 11/255
  NSAP address: 47.0091810000000002F26D4901.000011116666.06
  Encapsulation ATM
  292553397 packets input, -386762809 bytes
  164906758 packets output, 1937663833 bytes
  0 OAM cells input, 0 OAM cells output, loopback not set
  Keepalive not supported
  Encapsulation(s):, PVC mode
  1024 maximum active VCs, 28 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
  UNI Version = 4.0, Link Side = user
  Last input 00:00:00, output 2d05h, output hang never
  Last clearing of "show interface" counters never
  Input queue: -1902/75/0 (size/max/drops); Total output drops: 205
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 67000 bits/sec, 273 packets/sec
  5 minute output rate 136000 bits/sec, 548 packets/sec
     76766014 packets input, 936995443 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

```
     367264676 packets output, 3261882795 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

- Enter the **show atm video-voice address** privileged EXEC command to display the ATM interface address and confirm the ILMI status (ILMI PVC is set up to enable SVC management). The ATM interface is assigned automatically with the **atm voice aesa** command. The following is a sample output:

```
Router# show atm video-voice address

nsap address                                      type         ilmi status
47.0091810000000002F26D4901.00107B4832E1.FE   VOICE_AAL5   Confirmed
47.0091810000000002F26D4901.00107B4832E1.C8   VIDEO_AAL1   Confirmed
```

Verifying the VoATM Connection

Verify that the voice connection is working by performing the following steps:

Step 1 Pick up the handset on a telephone connected to the configuration and verify that there is dial tone.

Step 2 Make a call from the local telephone to a configured dial peer to verify the connection.

Step 3 Check the validity of the dial-peer and voice-port configuration by performing the following tasks:

- If there are relatively few dial peers configured, use the **show dial-peer voice** command to verify that the data configured is correct.
- To show the status of the voice ports, use the **show voice port** command.
- To show the call status for all voice ports, use the **show voice call** command.
- To show the current status of all DSP voice channels, use the **show voice dsp** command.

Troubleshooting Tips

If a call does not connect, resolve the problem by performing the following tasks:

- Verify dial peer configuration by using the **show dial-peer voice** command on the local and remote concentrators.
- Verify that ATM interface 0 is up by using the **show interface** command.
- Ensure that the voice port, serial port, and controller T1 0 are set to **no shutdown**.

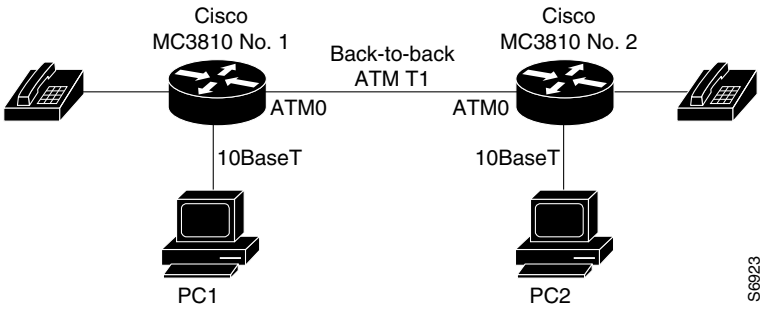
VoATM Configuration Examples

Configuration examples for VoATM are shown in the following sections:

Back-to-Back VoATM PVCs Example

Figure 106 shows a configuration example for two Cisco MC3810 multiservice concentrators configured back-to-back, with VoATM configured for both concentrators. This setup is a useful for testing the VoATM configuration locally to ensure that voice connections can be made before configuring VoATM across a larger network. Following the figure are the commands required for configuring the Cisco MC3810 multiservice concentrators in this example.

Figure 106 Back-to-Back VoATM PVCs Configuration



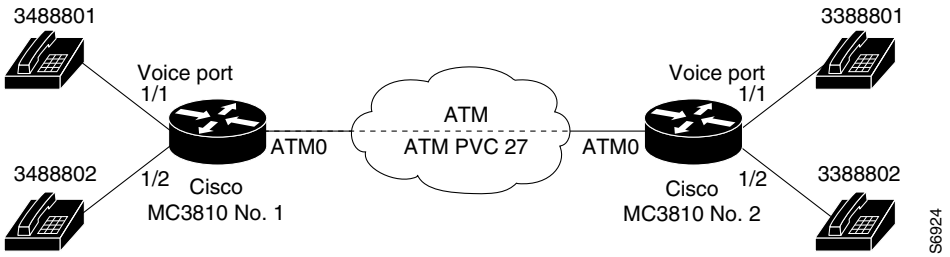
Cisco MC3810 Multiservice Concentrator 1	Cisco MC3810 Multiservice Concentrator 2
hostname location1 no ip domain-lookup ! interface Ethernet0 ip address 10.1.10.1 255.255.255.0 no ip mroute-cache no ip route-cache ! controller T1 0 clock source internal mode atm ! interface atm0 point-to-point ip address 10.1.1.1 255.255.255.0 no ip mroute-cache ! pvc 1 1 100 encapsulation aal5mux voice vbr-rt 384 192 48 !	hostname location2 no ip domain-lookup ! interface Ethernet0 ip address 10.1.20.1 255.255.255.0 no ip mroute-cache no ip route-cache ! controller T1 0 clock source line mode atm ! interface atm0 point-to-point ip address 10.1.1.2 255.255.255.0 no ip mroute-cache ! pvc 1 1 100 encapsulation aal5mux voice vbr-rt 384 192 48 !

Cisco MC3810 Multiservice Concentrator 1	Cisco MC3810 Multiservice Concentrator 2
pvc 2 1 200 encapsulation aal5snap map-group atm1 ! router rip redistribute connected network 10.0.0.0 ! no ip classless ! map-list atm1 ip 10.1.1.2 atm pvc 2 broadcast ! dial-peer voice 1 pots destination-pattern 10 port 1/1 ! dial-peer voice 202 voatm destination-pattern 2. session target ATM0 1 ! end	pvc 2 1 200 encapsulation aal5snap map-group atm1 ! router rip redistribute connected network 10.0.0.0 ! no ip classless ! map-list atm1 ip 10.1.1.1 atm pvc 2 broadcast ! dial-peer voice 1 pots destination-pattern 20 port 1/1 ! dial-peer voice 202 voatm destination-pattern 1. session target ATM0 1 ! end

Voice and Data Traffic over ATM PVCs Example

Figure 107 shows an example for both voice and data traffic over ATM between two Cisco MC3810 multiservice concentrators, including configuration for voice ports and dial peers. Following the figure are the commands required for configuring the Cisco MC3810 multiservice concentrators in this example.

Figure 107 Voice and Data Traffic over ATM PVCs Configuration



Cisco MC3810 Multiservice Concentrator 1	Cisco MC3810 Multiservice Concentrator 2
interface Ethernet0 ip address 172.22.124.239 255.255.0.0 ! controller T1 0 mode ATM ! interface atm0 point-to-point ip address 223.223.224.229 255.255.255.0 no ip mroute-cache no ip route-cache map-group atm1 ! pvc 26 26 200 encapsulation aal5snap ! pvc 27 27 270 encapsulation aal5mux voice vbr-rt 384 192 48 ! no ip classless ! map-list atm1 ip 223.223.224.228 atm pvc 26 broadcast ! voice-port 1/1 ! voice-port 1/2 ! voice-port 1/3 ! voice-port 1/4 ! dial-peer voice 1 pots destination-pattern 3488801 port 1/1 ! dial-peer voice 2 pots destination-pattern 3488802 port 1/2 ! end	interface Ethernet0 ip address 172.22.124.247 255.255.0.0 ! controller T1 0 mode ATM ! interface atm0 point-to-point ip address 223.223.224.228 255.255.255.0 no ip mroute-cache no ip route-cache map-group atm1 ! pvc 26 26 200 encapsulation aal5snap ! pvc 27 27 270 encapsulation aal5mux voice vbr-rt 384 192 48 ! no ip classless ! map-list atm1 ip 223.223.224.229 atm pvc 26 broadcast ! login line vty 1 4 login ! voice-port 1/1 ! voice-port 1/2 ! voice-port 1/3 ! voice-port 1/4 ! dial-peer voice 1 pots destination-pattern 3388801 port 1/1 ! dial-peer voice 2 pots destination-pattern 3388802 port 1/2 ! dial-peer voice 1001 voatm destination-pattern 348.... session target ATM0 27 ! end

VoATM for Cisco 3600 Series Routers Configuration Example

The following is a sample configuration for VoATM on a Cisco 3600 series router:

```
version 12.2
!
hostname c3640_1
!
no ip subnet-zero
no ip routing
ip wccp version 2
!
dial-control-mib max-size 500
!
process-max-time 200
!
interface Ethernet0/0
 ip address 172.28.129.54 255.255.255.192
 ip helper-address 171.71.20.62
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Serial0/0
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no fair-queue
!
interface Ethernet0/1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
!
interface ATM1/0
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 1/100
  vbr-rt 1000 500
  encapsulation aal5mux voice
 !
 no scrambling-payload
 impedance 120-ohm
 no fair-queue
!
interface ATM1/1
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 2/100
  vbr-rt 1000 500
  encapsulation aal5mux voice
 !
 no scrambling-payload
 impedance 120-ohm
```

```
 no fair-queue
!
interface ATM1/1.1 point-to-point
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 pvc 3/200
  vbr-rt 64 64 4
  encapsulation aal5mux voice
!
!
interface ATM1/2
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 no atm ilmi-keepalive
 no scrambling-payload
 impedance 120-ohm
 no fair-queue
!
interface ATM1/3
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 no atm ilmi-keepalive
 no scrambling-payload
 impedance 120-ohm
 no fair-queue
!
interface ATM1/4
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 no atm ilmi-keepalive
 no scrambling-payload
 impedance 120-ohm
 no fair-queue
!
interface ATM1/5
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 no atm ilmi-keepalive
 no scrambling-payload
 impedance 120-ohm
 no fair-queue
!
interface ATM1/6
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 no atm ilmi-keepalive
 no scrambling-payload
 impedance 120-ohm
```

```
 no fair-queue
!
interface ATM1/7
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 no atm ilmi-keepalive
 no scrambling-payload
 impedance 120-ohm
 no fair-queue
!
interface ATM3/0
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 map-group atm1
 atm clock INTERNAL
 pvc 2/200
  encapsulation aal5snap
 no atm auto-configuration
 no atm ilmi-keepalive
 no atm address-registration
 no atm ilmi-enable
 pvc voice 1/100
  vbr-rt 5000 2500
  encapsulation aal5mux voice
!
!
ip default-gateway 172.28.129.1
ip classless
ip route 171.70.20.62 255.255.255.255 172.28.129.1
no ip http server
!
!
map-list atm1
 ip 4.4.4.2 atm-vc 2 broadcast
!
map-class frame-relay fr1
!
map-class frame-relay voice
 no frame-relay adaptive-shaping
 frame-relay cir 128000
 frame-relay bc 128000
snmp-server engineID local 00000009020000107BC778C0
snmp-server community public RO
snmp-server community SNMPv2c view v2default RO
snmp-server community v2 view v1default RO
snmp-server community config view v1default RO
snmp-server community voice view v1default RO
snmp-server packetsize 4096
snmp-server enable traps snmp
snmp-server enable traps casa
snmp-server enable traps config
snmp-server enable traps voice poor-qov
snmp-server host 171.71.128.229 version 2c SNMPv2c  config voice snmp
snmp-server host 171.71.128.242 version 2c public config voice snmp
snmp-server host 171.71.129.16 version 2c public tty frame-relay isdn hsrp
config entity envmon bgp rsvp rtr syslog stun sdllc dspu rsrb dlsw sdlc snmp
snmp-server host 171.71.129.164 version 2c public  config voice snmp
!
```

```
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 session-timeout 10
 password apple
 login
!
voice-port 2/0/0
 input gain 5
 output attenuation 5
!
voice-port 2/0/1
 input gain 5
 output attenuation 5
!
voice-port 2/1/0
 input gain 5
 output attenuation 5
!
voice-port 2/1/1
 input gain 5
 output attenuation 5
!
dial-peer voice 2 pots
 destination-pattern 4001
!
dial-peer voice 8000 pots
 destination-pattern 84000
!
dial-peer voice 9000 pots
 destination-pattern 94000
!
dial-peer voice 9001 pots
 destination-pattern 94001
!
dial-peer voice 348 voatm
 destination-pattern 348....
 signal-type ext-signal
 session target ATM3/0 pvc 1/100
!
dial-peer voice 338 voatm
 destination-pattern 338....
 signal-type ext-signal
 session target ATM1/0 pvc 1/100
!
dial-peer voice 2222 voatm
 preference 1
 session target ATM1/0 pvc 1/100
!
dial-peer voice 9500 voatm
 destination-pattern 95...
 session target ATM3/0 pvc 1/100
!
dial-peer voice 8400 pots
 destination-pattern 84000
!
dial-peer voice 50000 voatm
 destination-pattern 5264000
 session target ATM3/0 pvc 1/100
!
dial-peer voice 10000 pots
 destination-pattern 5254000
```

```
 port 2/0/0
!
dial-peer voice 10001 pots
 destination-pattern 4000789
 port 2/1/0
!
num-exp 1 1234
num-exp 2 2234
num-exp 12 34567890
num-exp 55 66666
end
```

VoATM for the Cisco MC3810 Multiservice Concentrator Configuration Example

The following is a sample configuration for VoATM on Cisco MC3810 multiservice concentrators at opposite ends of an AAL2 trunk:

End A

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname aal2-faxtest1
!
network-clock base-rate 64k
ip subnet-zero
!
isdn voice-call-failure 0
!
voice-card 0
!
controller T1 0
 mode atm
 framing esf
 linecode b8zs
!
controller T1 1
 mode cas
 framing esf
 linecode b8zs
interface Ethernet0
 ip address 1.7.78.1 255.255.0.0
!
interface Serial0
 no ip address
!
interface Serial1
 no ip address
 shutdown
interface ATM0
 no ip address
 ip mroute-cache
 no atm ilmi-keepalive
 pvc 99/99
  vbr-rt 1536 1536 1000
  encapsulation aal2
!
voice-port 1:1
 no echo-cancel enable
```

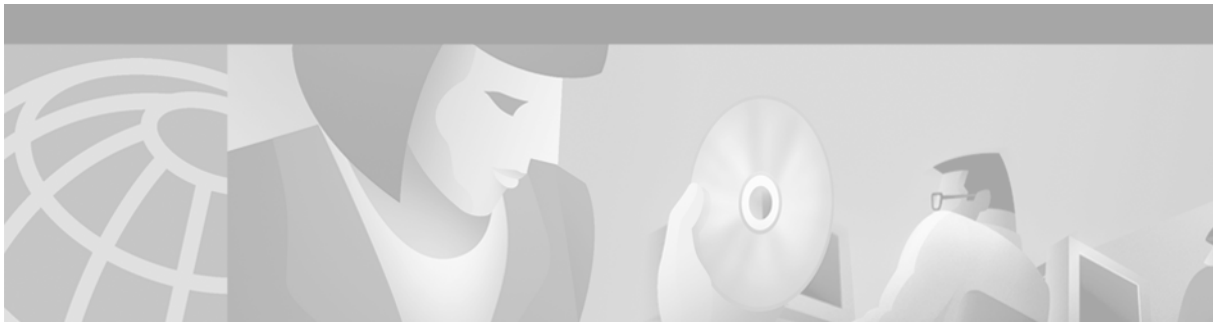
```
timeouts wait-release 3
 connection trunk 1001
!
dial-peer voice 1001 voatm
 destination-pattern 1001
 called-number 2001
session protocol aal2-trunk
 session target ATM0 pvc 99/99 21
 dtmf-relay
 signal-type transparent
 codec aal2-profile custom 100 g711ulaw
 no vad
!
dial-peer voice 201 pots
 destination-pattern 2001
port 1:1
end
```

End B

```
Current configuration:
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname aal2-faxtest2
!
network-clock base-rate 64k
ip subnet-zero
!
isdn voice-call-failure 0
!
voice-card 0
!
controller T1 0
 mode atm
 framing esf
 clock source internal
 linecode b8zs
!
controller T1 1
 mode cas
 framing esf
 linecode b8zs
 ds0-group 1 timeslots 1 type e&m-immediate-start

interface Ethernet0
 ip address 1.7.78.4 255.255.0.0
!
interface Serial0
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface ATM0
 ip address 223.223.226.3 255.255.255.0
 ip mroute-cache
 no atm ilmi-keepalive
 pvc 99/99
  vbr-rt 1536 1536 1000
```

```
  encapsulation aal2
!
voice-port 1:1
timeouts wait-release 3
 connection trunk 2001
!
dial-peer voice 201 pots
 destination-pattern 1001
port 1:1
!
dial-peer voice 1001 voatm
 destination-pattern 2001
 called-number 1001
 session protocol aal2-trunk
 session target ATM0 pvc 99/99 21
 dtmf-relay
 signal-type transparent
 codec aal2-profile custom 100 g711ulaw
 no vad
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line 2 3
line vty 0 4
 login
!
end
```

Telephony Applications



Configuring TCL IVR Applications

This chapter shows you how to configure Interactive Voice Response (IVR) using the Tool Command Language (TCL) scripts. This software release introduces TCL IVR Version 2.0 with several feature enhancements to the Cisco IVR functionality. This chapter contains the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

TCL IVR Overview

IVR consists of simple voice prompting and digit collection to gather caller information for authenticating the user and identifying the destination. IVR applications can be assigned to specific ports or invoked on the basis of DNIS. An IP public switched telephone network gateway can have several IVR applications to accommodate many different gateway services, and you can customize the IVR applications to present different interfaces to the various callers.

IVR systems provide information in the form of recorded messages over telephone lines in response to user input in the form of spoken words, or more commonly dual tone multifrequency (DTMF) signalling. For example, when a user makes a call with a debit card, an IVR application is used to prompt the caller to enter a specific type of information, such as an account number. After playing the voice prompt, the IVR application collects the predetermined number of touch tones and then places the call to the destination phone or system.

IVR uses TCL scripts gather information and to process accounting and billing. For example, a TCL IVR script plays when a caller receives a voice-prompt instruction to enter a specific type of information, such as a personal identification number (PIN). After playing the voice prompt, the TCL IVR application collects the predetermined number of touch tones and sends the collected information to an external server for user authentication and authorization.

TCL IVR Enhancements

Since the introduction of the Cisco IVR technology, the software has undergone several enhancements. TCL IVR Version 2.0 is made up of separate components that are described individually in the sections that follow. The enhancements are as follows:

- Media Gateway Control Protocol (MGCP) scripting package implementation
- Real Time Streaming Protocol (RTSP) client implementation
- TCL IVR prompt playout and digit collection on IP call legs
- New TCL verbs to utilize RTSP and MGCP scripting features

The enhancements add scalability and enable the TCL IVR scripting functionality on VoIP legs. In addition, support for RTSP enables VoIP gateways to play messages from RTSP-compliant announcement servers. The addition of these enhancements also reduces the CPU load and saves memory on the gateway because no packetization is involved. Larger prompts can be played, and the use of an external audio server is allowed.

Note TCL IVR 2.0 removed the signature locking mechanism requirement.

MGCP Scripting

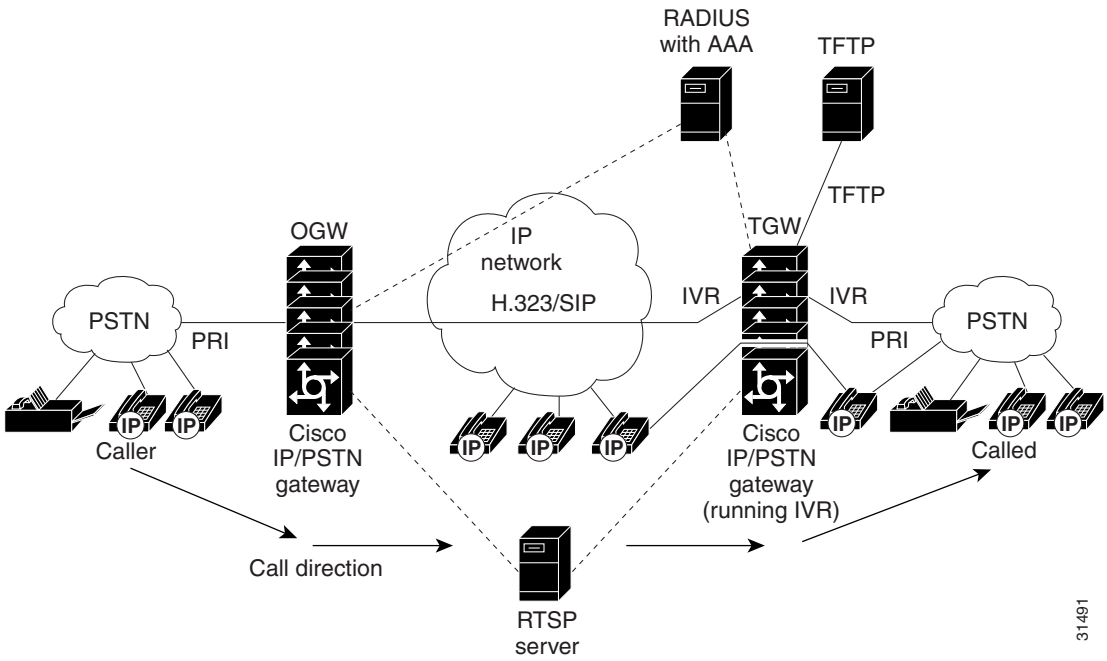
TCL IVR Version 2.0 infrastructure is greatly enhanced with the addition of support for MGCP using the application package model. MGCP defines application packages to run scripts on the media gateways. These application packages initiate scripts on the gateways and receive return values after execution completes. MGCP scripting allows external call agents (CAs) to instruct a media gateway to run an TCL IVR script in order to perform a specific task and return the end result. For example, you can request and collect the PIN and account number from a caller.

Two previously released Cisco VoIP features that can be implemented are the Debit Card for Packet Telephony and TCL IVR. Both features use the TCL scripting language. The TCL scripts that run with MGCP are written in TCL IVR API Version 2.0 and are able to receive calls through hand off. MGCP scripts can run any TCL command.

Note For more information about MGCP, see “Configuring Media Gateway Control Protocol and Related Protocols” chapter.

Figure 108 displays the CA controlling the TCL IVR scripts. MGCP is the protocol that is running on the CA. The RTSP server is configured to interact with the gateways that have TCL IVR scripts installed and running. The RADIUS server running authentication, authorization, and accounting (AAA) also interacts with the gateways.

Figure 108 MGCP Control of TCL IVR Scripts



RTSP Client Implementation

RTSP is an application-level protocol used for control over the delivery of data that has real-time properties. Using RTSP also enables an external RTSP server to play anouncements and interact with voice mail servers. It provides an extensive framework to enable control and to perform on-demand delivery of real-time data. For example, RTSP is used to control the delivery of audio streams from an audio server.

If you use an RTSP server in your network with VoIP gateways, a scripting application, (for example, an MGCP script) can run on the gateway and connect calls with audio streams from an external audio server. Using RTSP also has the following benefits:

- Reduces the CPU load
- Allows large prompts to be played that previously demanded high CPU usage from the gateway
- Saves memory on the gateway because no packetization is involved
- Allows use of an external audio server which removes the limitation on the number of prompts that can be played out and on the size of the prompt

TCL IVR Prompts Played on IP Call Legs

TCL IVR Version 2.0 scripts can be configured for incoming plain old telephone service (POTS) or VoIP call legs to play announcements to the user or collect user input (digits). With TCL IVR Version 2.0 the prompts can be triggered from both the PSTN side of the call leg and the IP side of the call leg. This feature enables the audio files (or prompts) to be played out over the IP network.

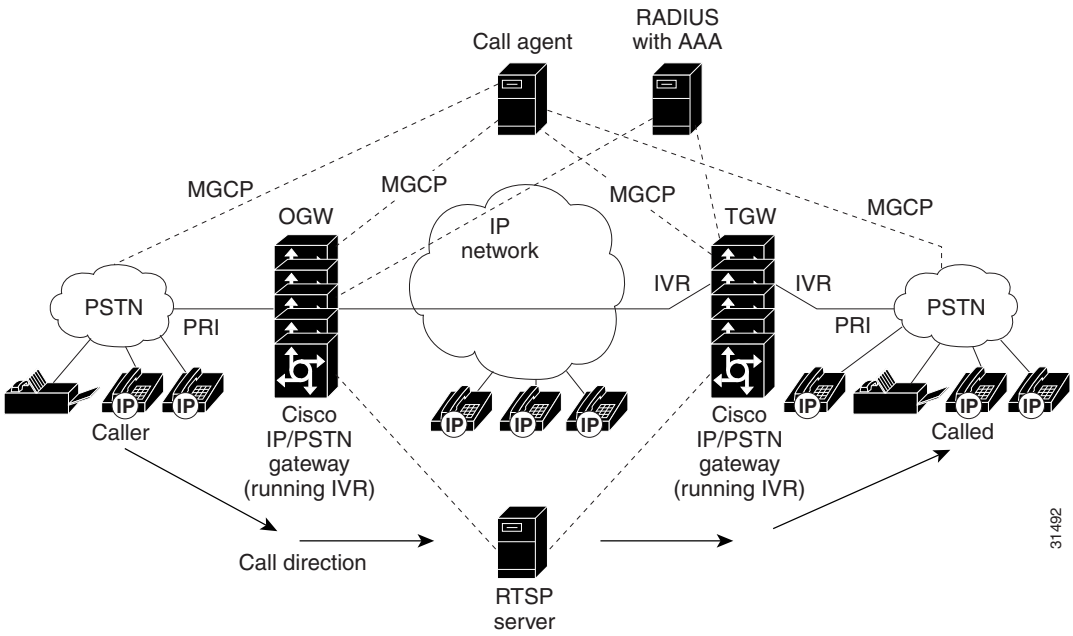
TCL IVR scripts played toward a VoIP call leg are subject to the following conditions:

- G.711 mu-law encoding must be used when prompts are played.
- G.711 mu-law encoding must also be used for the duration of these calls, even after prompt playout has completed.
- Digital signaling protocols (DSPs) can not be on the IP call leg so the script cannot initiate a tone.
- When an TCL IVR script is used to collect digits on a VoIP call leg, one of the following DTMF relay methods must be used.
  - For H.323 protocol configured on the call leg, use one of the following DTMF relay methods: Cisco proprietary RTP, H.245 Alphanumeric IE, or H.245 Signal IE
  - For SIP protocol configured on the call leg, use Cisco proprietary RTP

Note For additional information about the **dtmf-relay** command, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

IVR 2.0 enables the system to accept calls initiated from the IP side of the network using G.711, and terminate calls to the terminating gateway using the same codec. Figure 109 displays the TCL IVR application on the gateways controlling the scripts. IP phones can also originate a call to a gateway running an TCL IVR script.

Figure 109 IVR Control of Scripts on an IP Call Leg



TCL Verbs

TCL IVR, Version 2.0, delivers a new set of TCL verbs and scripts that replace the previous TCL version. The new TCL verbs enable the user to:

- Utilize the RTSP audio servers
- Develop TCL scripts that interact with the IVR application
- Pass events to the Media Gateway Controller, which is a call agent

TCL IVR Version 2.0 is not backward compatible with the IVR 1.0 scripts. The MGCP scripting package can only be implemented using the new TCL verbs.

Note For in-depth information about the TCL 2.0 verb set and how to develop scripts, refer to Cisco.com (Related Documentation index) and find the document, *TCL IVR API Version 2.0 Programmer's Guide*. The URL is:

http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/vapp_dev/index.htm.

The guide also contains an annotated example of a TCL IVR script and includes instructions for testing and loading TCL IVR scripts.

TCL IVR scripts use the TCL verbs to interact with the gateway during call processing in order to collect the required digits—for example, to request the PIN or account number for the caller. The TCL scripts are the default scripts for all Cisco voice features using IVR. TCL scripts are configured to control calls coming into or going out of the gateway.

Note Ensure that you have loaded the version of TCL scripts that support IVR Version 2. These TCL scripts can be downloaded from the following Cisco.com URL:

http://www.cisco.com/cgi-bin/tablebuild.pl/tclware.

The TCL IVR scripts shown below are listed as an example of the types of scripts available to be downloaded from the cisco.com Software Center. For a complete list of scripts, it is recommended that you check the Software Center.

Cisco provides the following IVR scripts:

- fax_hop_on_1—Collects digits from the redialer, such as account number and destination number. When a call is placed to an H.323 network, the set of fields (configured in the call information structure) are "entered", "destination", and "account".
- clid_authen—Authenticates the call with automatic number identification (ANI) and DNIS numbers, collects the destination data, and makes the call.
- clid_authen_npw—Performs as clid_authen, but uses a null password when authenticating, rather than DNIS numbers.
- clid_authen_collect—Authenticates the call with ANI and DNIS numbers and collects the destination data. If authentication fails, it collects the account and password.
- clid_authen_col_npw—Performs as clid_authen_collect, but uses a null password and does not use or collect DNIS numbers.

- clid_col_npw_3—Performs as clid_authen_col_npw except with that script, if authentication with the digits collected (account and PIN) fails, the clid_authen_col_npw script just plays a failure message (auth_failed.au) and then hangs up. The clid_col_npw_3 script allows two failures, then plays the retry audio file (auth_retry.au) and collects the account and PIN again.

  The caller can interrupt the message by entering digits for the account number, triggering the prompt to tell the caller to enter the PIN. If authentication fails the third time, the script plays the audio file auth_fail_final.au, and hangs up.

  Table 35 lists the prompt audio files associated with the clid_col_npw_3 script.

Table 35 clid_col_npw_3 Script Prompt Audio Files

Audio Filename	Action
flash:enter_account.au	Asks the caller to enter an account number. Played as the first request.
flash:auth_fail_retry.au	Asks the caller to reenter the account number. Plays after two failures.
flash:enter_pin.au	Asks the caller to enter a PIN.
flash:enter_destination.au	Asks the caller to enter a destination phone number.
flash:auth_fail_final.au	Informs the caller that the account number authorization has failed three times.

Table 36 lists additional audio files associated with the clid_col_npw_3 script.

Table 36 Additional clid_col_npw_3 Script Audio Files

Audio Filename	Action
auth_fail_retry.au	Informs the caller that authorization failed. Prompts the caller to reenter the account number followed by the pound sign (#).
auth_fail_final.au	Informs the caller, “I’m sorry, your account number cannot be verified. Please hang up and try again.”

- clid_col_npw_npw—Tries to authenticate by using ANI, null as the user ID, user, and user password pair. If that fails, it collects an account number and authenticates with account and null. It allows three tries for the caller to enter the account number before ending the call with the authentication failed audio file. If authentication succeeds, it plays a prompt to enter the destination number.

  Table 37 lists the audio files associated with the clid_col_npw_npw script.

Table 37 clid_col_npw_npw Script Audio Files

Audio Filename	Action
flash:enter_account.au	Asks the caller to enter the account number the first time.
flash:auth_fail_retry.au	Asks the caller to reenter the account number after first two failures.
flash:enter_destination.au	Asks the caller to enter the destination phone number.
flash:auth_fail_final.au	Informs the caller that the account number authorization has failed three times.

- clid_col_dnis_3.tcl—Authenticates the caller ID three times. First it authenticates the caller ID with DNIS. If that is not successful, it attempts to authenticate with the caller PIN up to three times.
- clid_col_npw_3.tcl—Authenticates with null. If authentication is not successful, it attempts to authenticate by using the caller PIN up to 3 times.
- clid_4digits_npw_3.tcl—Authenticates with null. If the authentication is not successful, it attempts to authenticate with the caller PIN up to 3 times using the 14-digit account number and password entered together.
- clid_4digits_npw_3_cli.tcl— Authenticates the account number and PIN respectively by using ANI and null. The number of digits allowed for the account number and password are configurable through the CLI. If the authentication fails, it allows the caller to retry. The retry number is also configured through the CLI.
- clid_authen_col_npw_cli.tcl—Authenticates the account number and PIN respectively using ANI and null. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.
- clid_authen_collect_cli.tcl—Authenticates the account number and PIN by using ANI and DNIS. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.
- clid_col_npw_3_cli.tcl—Authenticates by using ANI and null for account and PIN respectively. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI.
- clid_col_npw_npw_cli.tcl—Authenticates by using ANI and null for account and PIN respectively. If authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected together.

Note To display the contents of the TCL IVR script, use the **show call application voice** command.

TCL IVR Prerequisite Tasks

Before you configure your Cisco gateway to support TCL IVR, you must perform the following prerequisite tasks:

- Configure VoIP to support H.323-compliant gateways—meaning that in addition to the basic configuration tasks, such as configuring dial peers and voice ports, you must configure specific devices in your network to act as gateways.
- Configure a TFTP sever to perform storage and retrieval of the audio files, which are required by the Debit Card gateway or other features requiring TCL IVR scripts and audio files.
- Download the appropriate TCL IVR script from the Cisco.com. Use the **copy** command to copy your audio file (.au file) to your Flash memory, and the **audio-prompt load** command to read it into RAM. When you use TCL IVR applications, the gateway needs to know the URL where the TCL script can be found, as well as the URL of any audio file you want to use. Cisco IOS File System (IFS) is used to read the files, so any IFS-supported URLs can be used, which includes TFTP, FTP, or a pointer to a device on the router. During configuration of the application, you specify the URLs for the script and for the audio prompt. See the “Using URLs in IVR Scripts” chapter in the *TCL IVR API Version 2.0 Programmer's Guide* for more information.

- Make sure that your audio files are in the proper format. The TCL IVR prompts require audio file (.au) format of 8-bit, u-law, and 8-khz encoding. To encode your own audio files, we recommend that you use one of these two audio tools (or a tool of similar quality):
  - Cool Edit, manufactured by Syntrillium Software Corporation
  - AudioTool, manufactured by Sun Microsystems
- Make sure that your access platform has a minimum of 16 MB Flash and 128MB of DRAM memory.
- Install and configure the appropriate RADIUS security server in your network. The version of RADIUS that you are using must be able to support IETF-supported vendor specific attributes (VSAs), which are implemented by using IETF RADIUS attribute 26.

TCL IVR Configuration Tasks List

Before starting the software configuration tasks for the TCL IVR Version 2.0 features, complete the following preinstallation tasks:

- Download the TCL scripts and audio files to be used with this feature from the Cisco.com.
- Store the TCL scripts and audio files on a TFTP server configured to interact with your gateway access server.
- Create the TCL IVR application script to use with the **call application voice** command when configuring IVR using TCL scripts. You create this application first and store it on a server or location where it can be retrieved by the access server.
- Define the call flow and pass the defined parameter values to the application. Depending on the TCL script you select, these values can include the language of the audio file and the location of the audio file. Table 38 lists the TCL scripts and the parameter values they require.
- Associate the application to the incoming POTS or VoIP dial peer.

See the following sections for configuration tasks for the TCL IVR. Each task in the list is identified as either optional or required:

- Configuring the Call Application for the Dial Peer (Required)
- Configuring TCL IVR on the Inbound POTS Dial Peer or Configuring TCL IVR on the Inbound VoIP Dial Peer (Required)
- Configuring MGCP Scripting (Optional)

Note When an IVR script is used to detect a “long #” from a caller connected to the H.323 call leg, the DTMF method used must either be Cisco proprietary RTP or DTMF relay using H.245 signal IE. DTMF relay using H.245 alphanumeric IE does not report the actual duration of the digit, causing long pound (#) detection to fail.

Configuring the Call Application for the Dial Peer

You must configure the application that interacts with the dial peer before you configure the dial peer. The dial peer collects digits from the caller and uses the application you have created. Use the **call application voice** command as shown in the table that follows. Each command line is optional depending on the type of action desired or the digits to be collected.

To configure the application, enter the following commands in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# call application voice` *name url*	Defines the name of the application to be used with your TCL IVR script. The *url* argument specifies the location of the file and the access protocol. An example is as follows:
		`flash:scripts/session.tcl`
		`tftp://dirt/sarvi/scripts/session.tcl`
		`ftp://sarvi-ultra/scripts/session.tcl`
		`slot0:scripts/tcl/session..tcl`
		Note You can only configure a url if the application named *name* has *not* been configured.

Step 2	`Router(config)# call application voice` *name* `language` *digit language*	Specifies the language used by the audio files. An example is: call application voice test language 1 en. The arguments are as follows: • *digit*—Specifies zero (0) through 9. • *language*—Specifies two characters that represent a language. For example, “en” for English, “sp” for Spanish, and “ch” for Mandarin. Enter **aa** to represent all.
Step 3	`Router(config)# call application voice` *name* `pin-length` *number*	Defines the number of characters in the PIN for the designated application. Values are from 0 through 10.
Step 4	`Router(config)# call application voice` *name* `retry-count` *number*	Defines the number of times a caller is permitted to reenter the PIN for the designated application. Values are from 1 through 5.
Step 5	`Router(config)# call application voice` *name* `uid-length` *number*	Defines the number of characters allowed to be entered for the user ID for the designated application. Values are from 1 through 20.
Step 6	`Router(config)# call application voice` *name* `set-location` *language category location*	Defines the location, language, and category of the audio files for the designated application. An example is: **set-location** en 1 tftp://server dir/audio filename.

Table 38 lists TCL script names and the corresponding parameters that are required for each TCL scripts.

Table 38 TCL Scripts and Parameters

TCL Script Name	Description—Summary	Commands to Configure
clid_4digits_npw_3_cli.tcl	Authenticates the account number and PIN using ANI and null. The allowed length of digits is configurable through the CLI. If the authentication fails, it allows the caller to retry. The retry number is also configured through the CLI.	**call application voice uid-len** min = 1, max = 20, default - 10 **call application voice pin-len** min = 0, max - 10, default = 4 **call application voice retry-count** min = 1, max = 5, default = 3
clid_authen_col_npw_cli.tcl	Authenticates the account number and PIN using ANI and null. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.	**call application voice retry-count** min = 1, max = 5, default = 3
clid_authen_collect_cli.tcl	Authenticates the account number and PIN using ANI and DNIS. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected separately.	**call application voice retry-count** min = 1, max = 5, default = 3
clid_col_npw_3_cli.tcl	Authenticates using ANI and null for account and PIN. If the authentication fails, it allows the caller to retry. The retry number is configured through the CLI.	**call application voice retry-count** min = 1, max = 5, default = 3
clid_col_npw_npw_cli.tcl	Authenticates using ANI and null for account and PIN. If authentication fails, it allows the caller to retry. The retry number is configured through the CLI. The account number and PIN are collected together.	**call application voice retry-count** min = 1, max = 5, default = 3

Configuring TCL IVR on the Inbound POTS Dial Peer

Configuring gw-accounting and AAA are not always required for POTS dial peer configuration. It is dependent upon the type of application that is being used with TCL IVR. For example, the Pre-Paid Calling Card feature requires accounting and the authentication caller ID application does not.

To configure the inbound POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	(Optional) Enables AAA security and accounting services.
Step 2	`Router(config)# gw-accounting h323`	(Optional) Enables gateway-specific H.323 accounting.
Step 3	`Router(config)# aaa authentication login h323 radius`	(Optional) Defines a method list called H.323 where RADIUS is defined as the only method of login authentication.

	Command	Purpose
Step 4	`Router(config)# aaa accounting connection h323 start-stop radius`	(Optional) Defines a method list called H.323 where RADIUS is used to perform connection accounting, providing start-stop records.
Step 5	`Router(config)# radius-server host` *ip-address* `auth-port` *number* `acct-port` *number*	Identifies the RADIUS server and the ports that will be used for authentication and accounting services.
Step 6	`Router(config)# radius-server key` *key*	Specifies the password used between the gateway and the RADIUS server.
Step 7	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode to configure the incoming POTS dial peer. The *number* argument is a tag that uniquely identifies the dial peer.
Step 8	`Router(dial-peer)# application` *name*	Associates the TCL IVR application with the incoming POTS dial peer. Enter the selected TCL IVR application name.
Step 9	`Router(config-dial-peer)# destination-pattern` *string*	Enters the telephone number associated with this dial peer. The *pattern* argument is a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are numbers from zero (0) through nine and letters from A through D. The following special characters can be entered in the string: • Plus sign (**+**)—(Optional) Indicates an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Specifies the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – Asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. These characters cannot be used as leading characters in a string (for example, *650). – Comma (,) inserts a pause between digits. – Period (.) matches any entered digit (this character is used as a wildcard). The period cannot be used as a leading character in a string (for example, .650). • **T**—(Optional) Indicates that the destination-pattern value is a variable length dial-string.
Step 10	`Router(config-dial-peer)# session target`	Specifies the session target IP address.

Configuring TCL IVR on the Inbound VoIP Dial Peer

To configure the inbound VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice 4401 voip`	Enters the dial-peer configuration mode and identifies the call leg.
Step 2	`Router(config-dial-peer)# application application-name`	Specifies the name of the application and script to use.
Step 3	`Router(config-dial-peer)# destination-pattern pattern`	Enters the destination pattern.
Step 4	`Router(config-dial-peer)# session protocol sipv2`	Specifies the session protocol. The default session protocol is H.323. The *sipv2* argument enables SIP.
Step 5	`Router(config-dial-peer)# session target`	Specifies the session target IP address.
Step 6	`Router(config-dial-peer)# dtmf-relay cisco-rtp`	Specifies the DTMF relay method. The keyword **cisco-rtp** specifies H.323 and SIP. Other keywords that are available only for H.323 are **h245-alphanumeric** and **h245-signal**. **Note** If digit collection from this VoIP call leg is required, the command **dtmf-relay** is required. The default is **no dtmf-relay**.
Step 7	`Router(config-dial-peer)# codec g711ulaw`	Specifies the voice codec. **Note** If the configured application will be playing prompts to the VoIP call leg, the **g711ulaw** keyword is required.

Configuring MGCP Scripting

To perform MGCP scripting, you must enable the MGCP script package. Enable the script in global configuration mode by entering the **mgcp package-capability script package** command. The example MGCP configuration shown in this section is for DS0s on T1 lines. The configuration tasks are as follows:

- Enabling the MGCP service on the DS0 groups
- Enabling the other MGCP packages
- Configuring the call agent address and other MGCP parameters

To configure MGCP scripting, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mgcp`	Starts the MGCP daemon.
Step 2	`Router(config)# mgcp request timeout timeout`	Specifies how long the gateway should wait for a response to a request.
Step 3	`Router(config)# mgcp request retries count`	Specifies the number of times to retry sending the **mgcp** command.

	Command	Purpose
Step 4	`Router(config)# mgcp call-agent {ipaddr | hostname} [port]`	Configures the address of the call agent.
Step 5	`Router(config)# mgcp max-waiting-delay value`	Configures the maximum waiting delay to be used in a restart in progress (RSIP) message as restart instructions for the call agent.
Step 6	`Router(config)# mgcp restart-delay value`	Configures the restart delay value to be used in an RSIP message as graceful teardown instructions for the gateway connection.
Step 7	`Router(config)# mgcp vad`	Configure voice activity detection.
Step 8	`Router(config)# mgcp package-capability {as-package | dtmf-package | gm-package | rtp-package | trunk-package}`	Specifies an MGCP package capability.
Step 9	`Router(config)# mgcp default-package {as-package | dtmf-package | gm-package | rtp-package | trunk-package}`	Configures the default package capability type.
Step 10	`Router(config)# mgcp quality-threshold {hwm-jitter-buffer value | hwm-latency value | hwm-packet-loss value | lwm-jitter-buffer value | lwm-latency value | lwm-packet-loss value}`	Configures the jitter buffer size, packet-loss threshold, and latency threshold.
Step 11	`Router(config)# mgcp playout {adaptive init-value min-value max-value} | {fixed init-value}`	Tunes the jitter buffer packet size used for MGCP connections.
Step 12	`Router(config)# mgcp codec type [packetization-period value]`	Configures the default codec type.
Step 13	`Router(config)# mgcp ip-tos {high-reliability | high-throughput | low-cost | low-delay | precedence value}`	Enables the IP type of service for MGCP connections.
Step 14	`Router(config)# controller t1 slot#`	Uses the controller configuration mode for the T1 controller in the specified slot.
Step 15	`Router(config-controller)# framing type`	Configures the framing type.
Step 16	`Router(config-controller)# clock source type`	Configures the clock source.
Step 17	`Router(config-controller)# linecode type`	Configures the line coding.
Step 18	`Router(config-controller)# ds0-group n timeslots range type signaling-type service mgcp`	Configures the DS0s to support MGCP.

Verifying TCL IVR Configuration

You can verify TCL IVR configuration by performing the following tasks:

- To verify TCL IVR configuration parameters, use the **show running-config** command.
- To display a list of all voice applications, use the **show call application summary** command.
- To show the contents of the script configured, use the **show call application voice** command.
- To verify that the operational status of the dial peer, use the **show dial-peer voice** command.

To verify the TCL IVR configuration, perform the following steps:

Step 1 Enter the **show call application voice summary** command to verify that the newly created applications are listed. The example output follows:

```
Router# show call application voice summary

name                 description

DEFAULT              NEW::Basic app to do DID, or supply dialtone.
fax_hop_on           Script to talk to a fax redialer
clid_authen          Authenticate with (ani, dnis)
clid_authen_collect  Authenticate with (ani, dnis), collect if that fails
clid_authen_npw      Authenticate with (ani, NULL)
clid_authen_col_npw  Authenticate with (ani, NULL), collect if that fails
clid_col_npw_3       Authenticate with (ani, NULL), and 3 tries collecting
clid_col_npw_npw     Authenticate with (ani, NULL) and 3 tries without pw
SESSION              Default system session application
hotwo                tftp://hostname/scripts/nb/nb_handoffTwoLegs.tcl
hoone                tftp://hostname/scripts/nb/nb_dohandoff.tcl
hodest               tftp://hostname/scripts/nb/nb_handoff.tcl
clid                 tftp://hostname/scripts/tcl_ivr/clid_authen_collect.tcl
db102                tftp://hostname/scripts/1.02/debitcard.tcl
*hw                  tftp://171.69.184.xxx/tr_hello.tcl
*hw1                 tftp://san*tr_db
tftp://171.69.184.235/tr_debitcard.answer.tcl

TCL Script Version 2.0 supported.
TCL Script Version 1.1 supported.
```

Note In the output shown, an asterisk (*) in an application indicates that this application was not loaded successfully. Use the **show call application voice** command with the *name* argument to view information for a particular application.

Step 2 Enter the **show dial-peer voice** command with the *peer tag* argument and verify that the application associated with the dial peer is correct.

Step 3 Enter the **show running-config** command to display the entire configuration.

TCL IVR Configuration Examples

Use the **show running-config** command to display the entire gateway configuration. Figure 110 shows the type of topology used in the configuration for the example.

Figure 110 Example Configuration Topology

In this example configuration, GW1 is running TCL IVR for phone A, and GW2 is running TCL IVR for phone B.

This section provides the following configuration examples:

- TCL IVR for Gateway1 (GW1) Configuration Example, page 507
- TCL IVR for GW2 Configuration Example, page 510
- MGCP Scripting Configuration Example, page 512

TCL IVR for Gateway1 (GW1) Configuration Example

The following output is the result of using the **show running-config** command:

```
GW1
Router# show running-config

Building configuration...

Current configuration:

! Last configuration change at 08:39:29 PST Mon Jan 10 2000 by lab
!
version 12.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname GW1
!
logging buffered 100000 debugging
aaa new-model
aaa authentication login default local group radius
aaa authentication login h323 group radius
aaa authentication login con none
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
enable password xxx
!
username lab password 0 lab
!
resource-pool disable
!
clock timezone PST -8
ip subnet-zero
ip host baloo 1.14.124.xxx
ip host dirt 223.255.254.254
```

```
ip host rtspserver3 1.14.1xx.2
ip host rtspserver1 1.14.1xx.2
!
mgcp package-capability trunk-package
mgcp default-package trunk-package
isdn switch-type primary-net5
isdn voice-call-failure 0
!
tftp://dirt/hostname/WV/en_new/
call application voice debit_card tftp://dirt/Router/scripts.new/app_debitcard.tcl
call application voice debit_card uid-len 6
call application voice debit_card language 1 en
call application voice debit_card language 2 ch
call application voice debit_card set-location ch 0 tftp://dirt/hostname/WV/ch_new/
call application voice debit_card set-location en 0 tftp://dirt/hostname/WV/en_new/
call application voice debit_card_rtsp tftp://dirt/IVR 2.0/scripts.new/app_debitcard.tcl
call application voice debit_card_rtsp uid-len 6
call application voice debit_card_rtsp language 1 en
call application voice debit_card_rtsp language 2 ch
call application voice debit_card_rtsp set-location ch 0 rtsp://rtspserver1:554/
call application voice debit_card_rtsp set-location en 0 rtsp://rtspserver1:554/

mta receive maximum-recipients 0
!
controller E1 0
 clock source line primary
 pri-group timeslots 1-31
!
controller E1 1
!
controller E1 2
!
controller E1 3
!
gw-accounting h323
gw-accounting h323 vsa
gw-accounting voip
!
interface Ethernet0
 ip address 1.14.128.35 255.255.255.xxx
 no ip directed-broadcast
 h323-gateway voip interface
 h323-gateway voip id gk1 ipaddr 1.14.128.19 1xxx
 h323-gateway voip h323-id gw1@cisco.com
 h323-gateway voip tech-prefix 5#
!
interface Serial0:15
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-net5
     isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 16.0.0.1 255.255.xxx.0
 no ip directed-broadcast
 duplex full
 speed auto
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.14.128.33
ip route 1.14.xxx.0 255.xxx.255.xxx 16.0.0.2
```

```
ip route 1.14.xxx.16 255.xxx.255.240 1.14.xxx.33
no ip http server
!
radius-server host 1.14.132.2 auth-port 1645 acct-port 1646
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
!
voice-port 0:D
 cptone DE
!
dial-peer voice 200 voip
 incoming called-number 53
 destination-pattern 34.....
 session target ipv4:16.0.0.2
 dtmf-relay h245-alphanumeric
 codec g711ulaw
!
dial-peer voice 102 pots
 application debit_card_rtsp
 incoming called-number 3450072
 shutdown
 destination-pattern 53.....
 port 0:D
!
dial-peer voice 202 voip
 shutdown
 destination-pattern 34.....
 session protocol sipv2
 session target ipv4:16.0.0.2
 dtmf-relay cisco-rtp
 codec g711ulaw
!
dial-peer voice 101 pots
 application debit_card
 incoming called-number 3450070
 destination-pattern 53.....
 port 0:D
!
gateway
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password xxx
!
ntp clock-period 17180740
ntp server 1.14.42.23
end

GW1#
```

TCL IVR for GW2 Configuration Example

The following output is the result of using the **show running-config** command:

```
GW2#
Router# show running-config

Building configuration...

Current configuration:
!
! Last configuration change at 08:41:12 PST Mon Jan 10 2000 by lab
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname GW2
!
logging buffered 100000 debugging
aaa new-model
aaa authentication login default local group radius
aaa authentication login h323 group radius
aaa authentication login con none
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
!
username lab password xxx
username 111119 password xxx
!
resource-pool disable
!
clock timezone PST -8
ip subnet-zero
ip host radiusserver2 1.14.132.2
ip host radiusserver1 1.14.138.11
ip host baloo 1.14.124.254
ip host rtspserver2 1.14.136.2
ip host dirt 223.255.254.254
ip host rtspserver3 1.14.126.2
!
mgcp package-capability trunk-package
mgcp default-package trunk-package
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
call application voice clid_authen_sky
tftp://dirt/hostname/sky_scripts/clid_authen_collect_cli_sky.tcl

call application voice rtsp_demo tftp://dirt/hostname/sky_scripts/rtsp_demo.tcl
tftp://dirt/hostname/WV/en_new/
call application voice debit_card tftp://dirt/IVR 2.0/scripts.new/app_debitcard.tcl
call application voice debit_card uid-len 6
call application voice debit_card language 1 en
call application voice debit_card language 2 ch
call application voice debit_card set-location ch 0 tftp://dirt/hostname/WV/ch_new/
call application voice debit_card set-location en 0 tftp://dirt/hostname/WV/en_new/
call application voice clid_authen_rtsp tftp://dirt/IVR
2.0/scripts.new/app_clid_authen_collect_cli_rtsp.tcl

call application voice clid_authen_rtsp location rtsp://rtspserver2:554/
```

```
call application voice clid_authen1 tftp://dirt/IVR
2.0/scripts.new/app_clid_authen_collect_cli_rtsp.tcl
call application voice clid_authen1 location tftp://dirt/hostname/WV/en_new/
call application voice clid_authen1 uid-len 6
call application voice clid_authen1 retry-count 4
mta receive maximum-recipients 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 clock source line secondary 1
!
controller T1 2
!
controller T1 3
!
gw-accounting h323
gw-accounting h323 vsa
gw-accounting voip
!
interface Ethernet0
 ip address 1.14.xxx.4 255.255.xxx.240
 no ip directed-broadcast
 h323-gateway voip interface
 h323-gateway voip id gk2 ipaddr 1.14.xxx.18 1719
 h323-gateway voip h323-id gw2@cisco.com
 h323-gateway voip tech-prefix 3#
!
interface Serial0:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 16.0.0.2 255.xxx.255.0
 no ip directed-broadcast
 duplex full
 speed 10
 no cdp enable
!
ip classless
ip route 0.0.0.0 0.0.0.0 1.14.xxx.5
ip route 1.14.xxx.32 255.255.xxx.240 16.0.0.1
no ip http server
!
radius-server host 1.14.132.2 auth-port 1645 acct-port 1646
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication
!
voice-port 0:D
!
dial-peer voice 100 voip
 application debit_card
 incoming called-number 34
 shutdown
 destination-pattern 53.....
```

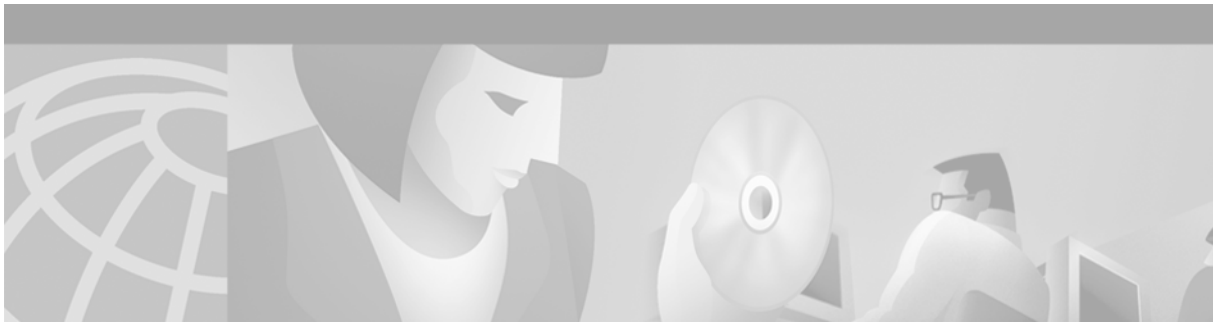
```
 session target ras
 dtmf-relay h245-alphanumeric
 codec g711ulaw
!
dial-peer voice 200 pots
 incoming called-number 30001
 destination-pattern 3450070
 port 0:D
 prefix 50070
!
dial-peer voice 101 voip
 application debit_card
incoming called-number 34.....
 shutdown
 session protocol sipv2
 session target ipv4:16.0.0.1
 dtmf-relay cisco-rtp
 codec g711ulaw
!
dial-peer voice 102 voip
 incoming called-number 34.....
 destination-pattern 53.....
 session target ipv4:16.0.0.1
 dtmf-relay h245-alphanumeric
 codec g711ulaw
!
gateway
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password xxx
!
ntp clock-period 17180933
ntp server 1.14.42.23
end

GW2#
```

MGCP Scripting Configuration Example

The following example displays only the MGCP specific portion of the configuration:

```
!
mgcp
mgcp request timeout 10000
mgcp request retries 1
mgcp call-agent 1.14.138.11
mgcp restart-delay 10
mgcp codec g723ar63 packetization-period 30
mgcp vad
mgcp package-capability gm-package
mgcp package-capability dtmf-package
mgcp package-capability trunk-package
mgcp package-capability rtp-package
mgcp package-capability as-package
mgcp package-capability script-package
mgcp default-package trunk-package
isdn switch-type primary-5ess
```

```
isdn voice-call-failure 0
!
mta receive maximum-recipients 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 1
 framing esf
 clock source line secondary 1
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 2
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
controller T1 3
 framing esf linecode b8zs
 ds0-group 0 timeslots 1-24 type none service mgcp
!
end
```



Configuring Debit Card Applications

This chapter explains how to configure debit cards for packet telephony and contains the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Debit Card for Packet Telephony Overview

The debit card for packet telephony application works in conjunction with the Cisco Interactive Voice Response (IVR) software, authentication, authorization, and accounting (AAA), RADIUS, and an integrated third-party billing system. The IVR software infrastructure allows prerecorded audio files to be combined dynamically to play the dollar amount of credit remaining on a customer's debit card, the time and date, and other information. For more information regarding Tool Command Language (TCL) and IVR, see the "Configuring TCL IVR Applications" chapter.

The integrated third-party billing system maintains per-user credit balance information. The AAA and RADIUS (vendor-specific attribute) VSAs communicate per-user credit balance information using the the billing system. The billing system and Cisco IOS software enable a carrier to authorize voice calls and debit individual user accounts in real time at the edges of a Voice over IP (VoIP) network without requiring external service nodes.

Note The Debit Card for Packet Telephony feature functionality is dependent upon the working configuration of the designated RADIUS server, which can be different servers controlling different VSA attributes.

The Debit Card for Packet Telephony feature provides the following functionality:

- Rates a call according to the caller ID, personal identification number (PIN), and destination number.

  The call is authenticated using caller ID and PIN. The RADIUS server provides the caller with a credit (dollar) amount. The caller is then prompted to enter the destination number. The TCL script authorizes the call with the RADIUS server. The RADIUS server keeps track of the caller credit information and it communicates with billing servers, if necessary, to maintain the credit information.

- Plays the credit (dollar amount) remaining on a card in $$$$$$.$$ format.

  The RADIUS server maintains the credit information and furnishes it to the script at the time of authentication. The IVR TCL script plays a prompt that announces the remaining credit to the caller as an amount in dollars and cents. The design is flexible enough to play any amount up to a maximum of $999999.99.

- Announces the "time-remaining" credit on the card in hours and minutes (HH:MM).

  The RADIUS server provides the remaining credit during the authorization phase. The TCL script combines prerecorded audio files to form the final prompt and play the "time-remaining" message to the caller. The time credit amount returned by the RADIUS server takes into consideration the rating and time boundary overlaps. The prompt played to the caller is, for example, "You have 5 hours and 35 minutes." The design is flexible enough to play any amount of time up to a maximum time specified in the script.

- Plays a "time-running-out" message based on the configurable time-out value.

  The RADIUS server maintains and furnishes credit information during the authorization phase. The IVR TCL script monitors the time remaining and, based on the configured value, plays a "time-has-run-out" message to the caller. The called party hears silence during this time. For example, if the time-out value is configured for 3 minutes, the prompt "You only have 3 minutes remaining on your credit" is played.

- Plays a warning "time-has-run-out" message when the credit runs out.

  This message is played to the calling party by the TCL script when the time credit has run out. The called party hears silence. The message is, for example, "Sorry, you have run out of credit."

- Makes more than one successive call to different destinations during a single call session.

  The Debit Card for Packet Telephony feature makes it possible for the caller to make subsequent calls to different destinations without disconnecting from the call leg. The caller is required to enter the account ID number and PIN only once during initial authorization. To make subsequent calls, the caller needs to enter only the destination number.

  After reaching one destination, the caller is allowed to disconnect the call by pressing the pound (#) key on the keypad and holding it down for 1 to 2 seconds. If the # key is pressed down for more than 1 second, it is treated as a long pound. The called party is disconnected and an announcement is played to the caller, giving the time remaining, and prompting for a new destination number.

  This feature also allows the caller to make additional calls if the called party hangs up.

- Reauthorizes each new call.

  Every time a caller enters a new destination number, the IVR TCL script reauthorizes the call with the RADIUS server and obtains the remaining time and credit balance information. The IVR TCL script then announces the amount of time remaining to the calling party.

- Allows type-ahead keypad entries without waiting for the prompt to complete.

The normal terminating character for the caller ID, PIN, and destination number is the pound (#) key. The caller may want to continue without waiting to hear the prompts. This TCL script will stop playing or will refrain from starting a prompt when it discovers that the caller wants to type ahead.

- Allows the caller to skip past announcements by pressing a touch-tone key.

  This IVR TCL script stops playing announcements when the system determines that the caller has pressed any touch-tone key.

- Allows retry when entering data (user ID, PIN, destination number) by using a special key.

  The caller is allowed to interrupt partially entered numbers and restart from the beginning by pressing the asterisk (*) key on the keypad. The asterisk key is configured in the IVR TCL script. The caller can use the asterisk key to cancel an entry and reenter the user ID, PIN, destination number. The caller is allowed to reenter data only a certain number of times. The number of retries is configurable; the default is three.

- Terminates a field by size rather than by using the terminating character (#).

  The IVR TCL script can be used to specify a number of digits in the user ID and PIN fields—meaning that the caller can enter all the digits (without the terminating character) and the script determines how to extract different fields from the number strings. If the caller uses the terminating character (the # key), the terminating character takes precedence and the fields are extracted accordingly.

- Supports two languages.

  The language is selected when the caller presses a predefined key. For example, “For English, press 1. For Mandarin, press 2.” The IVR TCL script uses the selected language until the caller disconnects. The caller is asked only once, at the beginning of the session, for the language of choice. In addition, the Debit Card for Packet Telephony feature determines how many languages are configured and plays the language selection menu only if needed.

- Sends an off-net tone to the caller.

  The RADIUS server maintains information regarding off-net calls. During authorization, it provides this information to the IVR TCL script. Based on the collected information, the IVR TCL script has the ability to generate a prerecorded message or tone to the calling party.

- Provides voice-quality information to the RADIUS server on a call-by-call basis.

  A new field has been added to the Stop Record field. The data for this field is obtained from fields that maintain and tune voice quality. It is the responsibility of the user application on the RADIUS server to use this information and give credit to the caller if the call has unsatisfactory voice quality.

- Uses prompt memory more efficiently.

  When voice prompts are not used for a period of time, they are swapped out of RAM. The swapping does not introduce undue delays in playing prompts. The most frequently used prompts remain in memory and are not swapped.

- Creates dynamic prompts by using prerecorded audio files.

  The Debit Card for Packet Telephony feature provides a general infrastructure that allows concatenating prerecorded audio files to play the dollar amount, time, and day. An interface for the scripts to use this infrastructure is also part of this feature. Dynamic creation of the final audio (by concatenating prerecorded audio files) is limited to playing out dollar amount, time, and day information. For example, when the system receives a credit balance of $15.50, it concatenates the prerecorded audio files, “You have” “15” “dollars” “and” “50” “cents” to make up this message.

- Allows retries for RADIUS server failures, with the maximum number of retries allowed determined by the RADIUS server.

If errors are returned by the RADIUS server during authentication or authorization (by use of the AAA application), the caller is allowed to retry the entry. The RADIUS server determines how many retries to allow. The caller is disconnected when the number of retries has exceeded the limit.

Debit Card Call Flow

A high-level call flow sequence is displayed in Figure 111 through Figure 115. The actual call flow varies, depending on the parameters passed to the application and on the features that are available on the RADIUS server billing system that is being used.

The call sequence flowcharts graphically depict the various states in the Debit Card for Packet Telephony application. The states are represented by the boxes with double bars and show the flow from one state to the next.

Figure 111 Debit Card Call Sequence 1

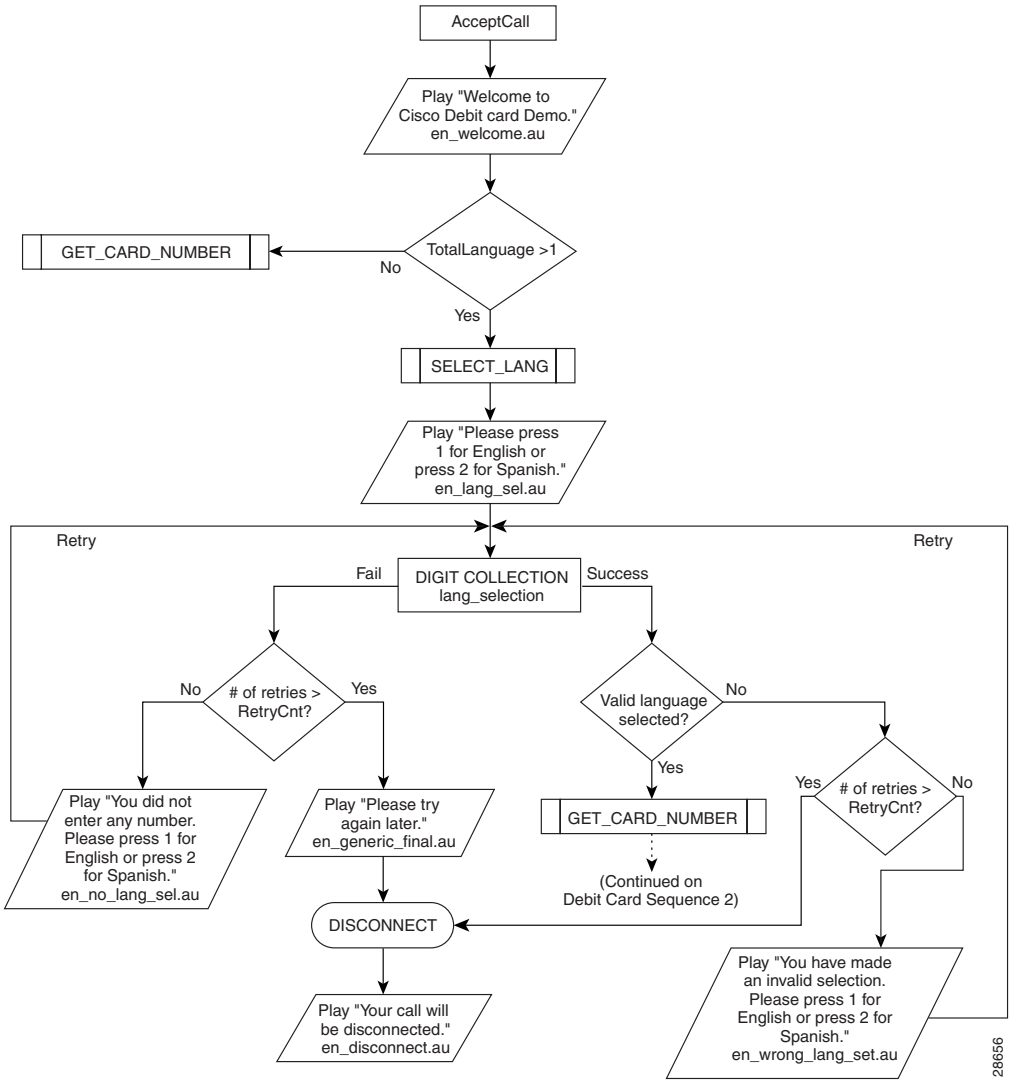


Figure 112 Debit Card Call Sequence 2

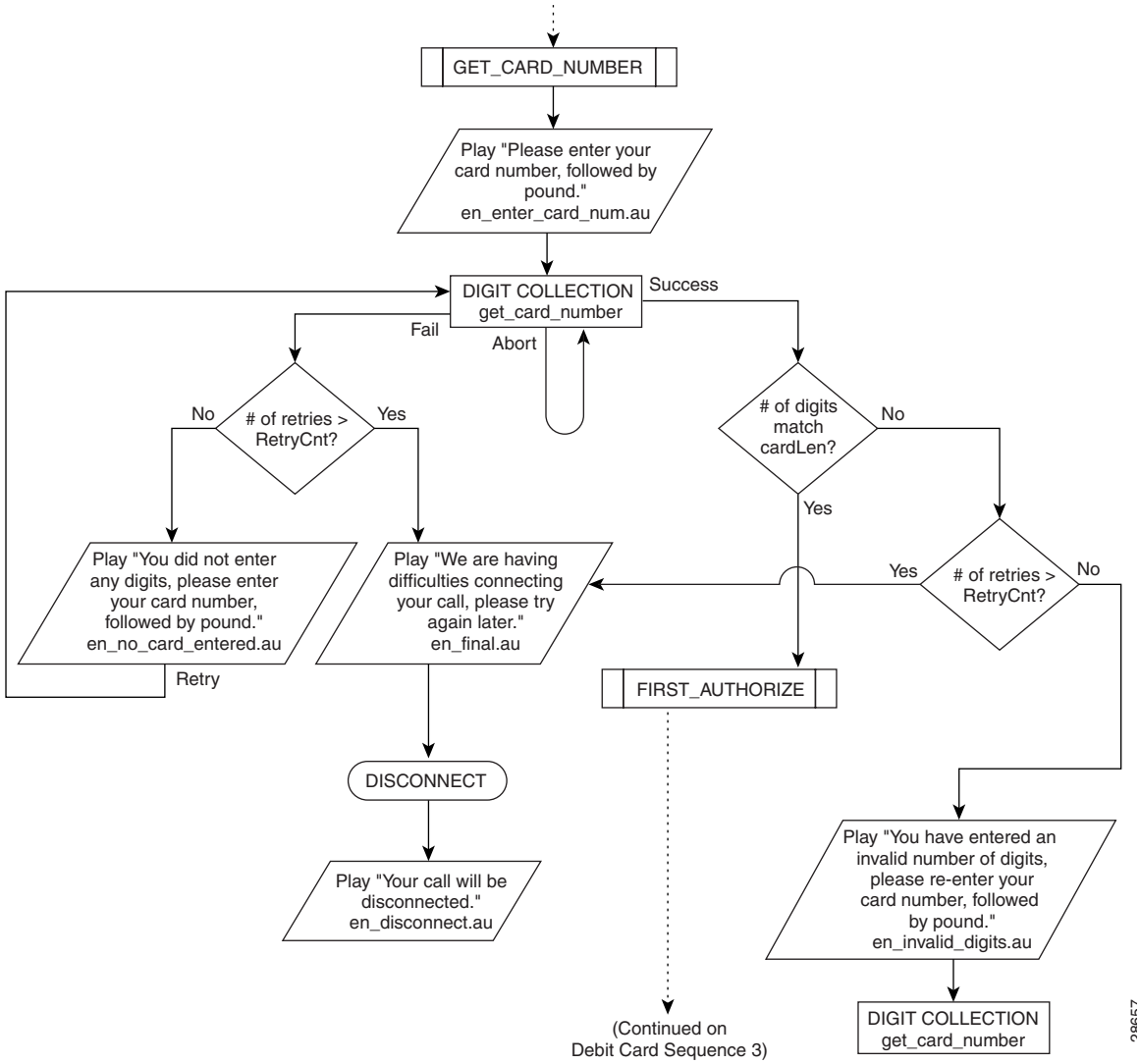


Figure 113 Debit Card Call Sequence 3

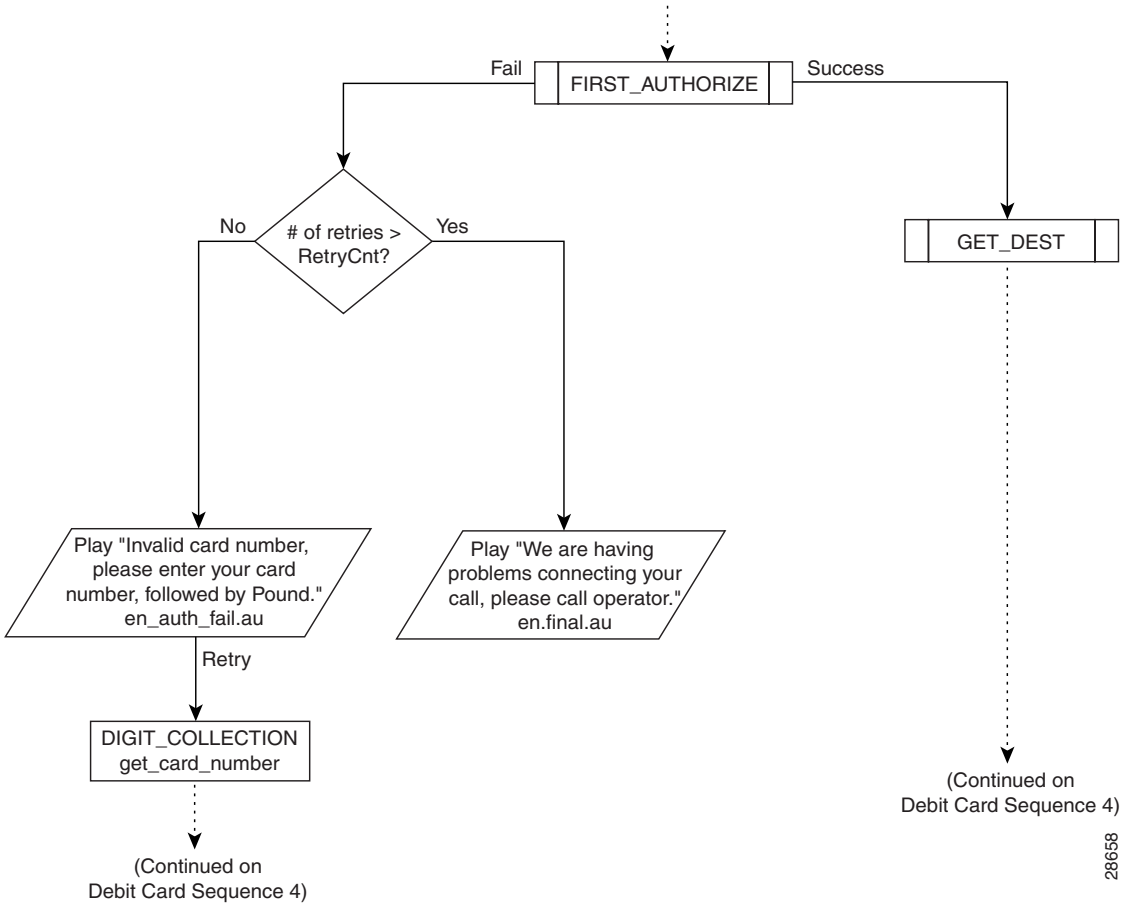


Figure 114 Debit Card Call Sequence 4

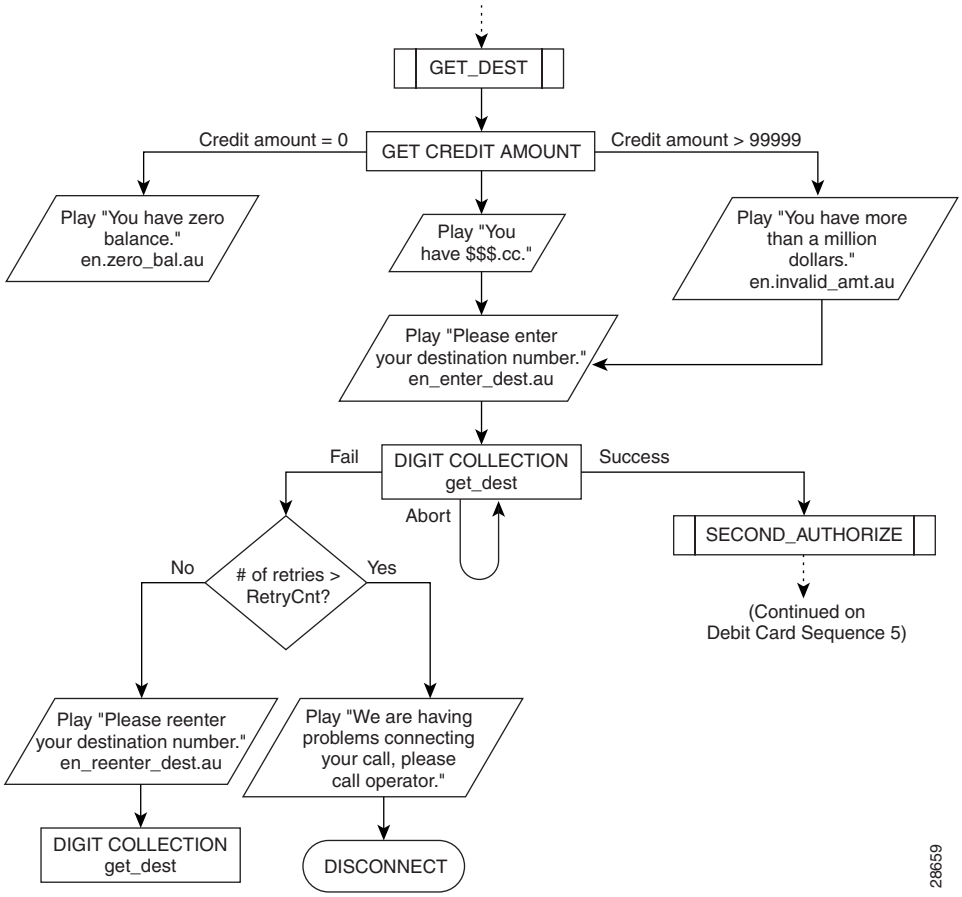
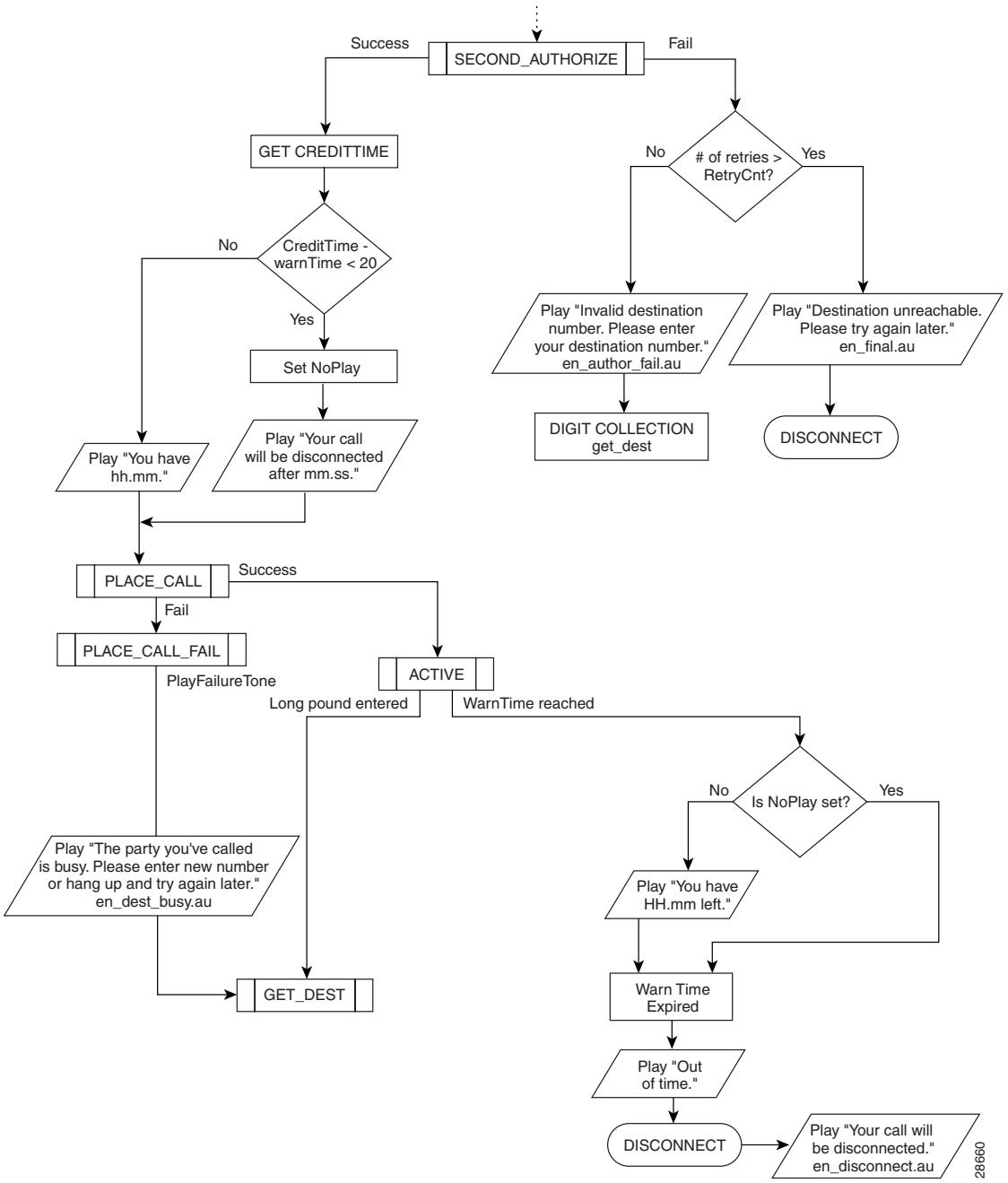


Figure 115 Debit Card Call Sequence 5



RADIUS and H.323 Gateway-Specific Accounting

A call leg is a discrete segment of a call connection that lies between two points in the connection. Each call made through a gateway has two call legs: incoming and outgoing. The RADIUS server collects VSAs during the accounting process for each call leg created on the gateway.

In order for the Debit Card for Packet Telephony feature to work with the RADIUS server to collect the appropriate connection accounting information, you must configure AAA to use H.323 gateway-specific accounting.

Audio File Prompts

Cisco provides a set of professionally recorded English (U.S.), Spanish, and Mandarin audio prompts to allow easier immediate integration and use of the debit card feature. The prompts are stored on an FTP server in the audio file format (.au). To announce the credit available to the caller, the system concatenates a series of prompts to create the appropriate message.

The Cisco-provided audio files are compressed and stored on Cisco.com. The audio files must be downloaded either to a TFTP server or into Flash memory. When the system runs, the files are copied into memory.

Note You can find TCLWare and audio files at the following URL: http://www.cisco.com/cgi-bin/tablebuild.pl/tclware

Because there is a large number of prompts, and to ensure efficient use of system memory resources, all prompts are downloaded to a TFTP server. A basic set of audio files is downloaded to the system when it is initialized. The system removes less frequently used prompts from memory to conserve memory. When a prompt is needed, the system retrieves the prompt from the TFTP server.

For languages that are syntactically similar to English, the audio file can be recorded and saved as the same filename so that the system can construct the message properly. It is not mandatory to use the prompt set that Cisco provides. If the customer has access to a recording studio, prompts can be created or customized as long as they are saved in the proper format.

Cisco-Provided Audio Files

The following audio file prompts are provided by Cisco. A similar set is available in Mandarin and in Spanish. The audio files provided by Cisco are listed in Table 39 and Table 40.

Table 39 Numbers Audio File Set

Audio Filename	Recorded Prompt	Audio Filename	Recorded Prompt
en_zero.au	Zero	en_fifteen.au	Fifteen
en_one.au	One	en_sixteen.au	Sixteen
en_two.au	Two	en_seventeen.au	Seventeen
en_three.au	Three	en_eighteen.au	Eighteen
en_four.au	Four	en_nineteen.au	Nineteen
en_five.au	Five	en_twenty.au	Twenty
en_six.au	Six	en_thirty.au	Thirty

Table 39 Numbers Audio File Set (continued)

Audio Filename	Recorded Prompt	Audio Filename	Recorded Prompt
en_seven.au	Seven	en_forty.au	Forty
en_eight.au	Eight	en_fifty.au	Fifty
en_nine.au	Nine	en_sixty.au	Sixty
en_ten.au	Ten	en_seventy.au	Seventy
en_eleven.au	Eleven	en_eighty.au	Eighty
en_twelve.au	Twelve	en_ninety.au	Ninety
en_thirteen.au	Thirteen	en_hundred.au	Hundred
en_fourteen.au	Fourteen	en_thousand.au	Thousand

Table 40 Miscellaneous Prompts

Audio Filename	Recorded Prompt
en_second.au	Second
en_seconds.au	Seconds
en_minute	Minute
en_minutes	Minutes
en_hour.au	Hour
en_hours.au	Hours
en_cent.au	Cent
en_cents.au	Cents
en_dollar.au	Dollar
en_dollars.au	Dollars

Additional Miscellaneous Prompts

The Debit Card for Packet Telephony feature provides the following additional miscellaneous prompts:

- en_welcome.au—"Welcome to Cisco Debit Card Demo."
- en_lang_select.au—"Please press 1 for English, 2 for Mandarin."
- en_wrong_lang_sel.au—"You have made an invalid selection. Please press 1 for English or press 2 for Mandarin."
- en_no_lang_sel.au—"You did not select any language. Press 1 for English or press 2 for Mandarin."
- en_final.au—"We are having difficulties connecting your call. Please try again later."
- en_generic_final.au—"Please hang up and try again."
- en_enter_card_num.au—"Please enter card number followed by the pound key."
- en_invalid_digits.au—"You have entered an invalid number of digits. Please reenter your card number followed by the pound key."
- en_auth_fail.au—"You have entered an invalid card number. Please reenter your card number followed by the pound key."

- en_no_card_entered.au—"You did not enter any digits. Please enter card number followed by the pound key."
- en_technical_problem.au—"We are having technical difficulties. Please call back later."
- en_zero_bal.au—"You have zero balance. Please call the operator or hang up."
- en_enter_dest.au—"Please enter destination number."
- en_disconnect.au—"Your call will be disconnected."
- en_disconnected.au—"You have been disconnected."
- en_dest_collect_fail.au—"Sorry, the number you have dialed is blocked. If you feel you have reached a number in error, please call the customer service number."
- en_invalid_amt.au—"You have more than one million."
- en_dest_busy.au—"The party you called is busy. Please enter a new number or hang up and try again later."
- en_enter_acct.au—"Please enter your account number followed by the pound key."
- en_no_acct_entered.au—"We did not get any input. Please enter your account number followed by the pound key."
- en_invalid_digits_acct.au—"You have entered an invalid number of digits. Please enter your account number followed by the pound key."
- en_invalid_account.au—"You have entered an invalid account number. Please enter your account number followed by the pound key."
- en_enter_pin.au—"Please enter your PIN number followed by the pound key."
- en_no_pin_entered.au—"We did not get any input. Please enter your PIN number followed by the pound key."
- en_invalid_digits_pin.au—"You have entered an invalid number of digits. Please enter your PIN number followed by the pound key."
- en_invalid_pin.au—"You have entered an invalid PIN. Please enter your PIN number followed by the pound key."
- en_card_expired.au—"We are sorry. Your card has expired."
- en_account_blocked.au—"This account is currently in use."
- en_no_dest_entered.au—"We did not get any input. Please enter the destination number you are calling."
- en_invalid_digits_pin.au—"You have entered an invalid number of digits. Please enter your PIN number followed by the pound key."
- en_invalid_pin.au—"You have entered an invalid PIN. Please enter your PIN number followed by the pound key."
- en_card_expired.au—"We are sorry. Your card has expired."
- en_account_blocked.au—"This account is currently in use."
- en_no_dest_entered.au—"We did not get any input. Please enter the destination number you are calling."
- en_no_dialpeer_match.au—"You have entered an invalid destination. Please reenter the destination number you are calling."
- en_connect_cust_ser.au—"You will be connected to Customer Service."
- en_dial_cust_ser.au—"Please hang up and dial the calling card customer service number."

- en_no_service.au—"We are sorry. This service is not available."
- en_dest_unreachable.au—"We are sorry. The destination you have called is unreachable."
- en_toll_free.au—"You can only make toll-free calls."

Audio Filenaming Convention

If you record your own audio files, you must name them using the convention described in Table 41 in order for the TCL scripts to identify which audio file to use. The TCL scripts are designed to work with designated audio filenames.

Table 41 Audio Filenaming Convention

Audio Filename	Description
en_one.au	Specifies the English language audio file for the number 1.
ch_one.au	Specifies the Mandarin language audio file for the number 1.
sp_one.au	Specifies the Spanish language audio file for the number 1.

For example, when the audio file for the caller to choose a language (en_lang_select.au) is played, ("Please press 1 for English, 2 for Mandarin."), if Mandarin is selected by the caller, then the TCL script calls the <ch> audio files to interact with the system.

Continuing with this example, when configuring the voice platform to process Mandarin audio files, use the **call application voice** command, with the **language** keyword and *set-location* argument. When specifying the *set-location* argument, you must configure the correct language-ch specifier to interact with the TCL script. Therefore, when naming your audio files, make sure you include the language identifier for each file.

Creating Audio Index Files

If you record your own audio files, you must also create an index file that contains a list of the audio files in URL format. An index file must be created for each audio file that is downloaded from TFTP to memory. Use the **ivr autoload** command to download the audio files into Flash memory.

When creating your audio file index, remember that the filename and extension (.au) are actually the URL of the file. Follow these recommendations:

- Each line should list only one file location (URL).
- Comment lines start with #.
- Locations listed in the index file should match the locations used by the application.
- Extra spaces at the beginning and end of the line are ignored.
- No spaces are allowed within the URL.

Sample Index File

The following is a sample index file:

```
# tftp://jurai/tclware/au/en/auth_fail_final.au
# tftp://jurai/tclware/au/en/auth_fail_retry.au

# tftp://jurai/tclware/au/en/auth_fail_retry_number.au
# tftp://jurai/tclware/au/en/auth_failed.au

# tftp://jurai/tclware/au/en/ch_generic_final.au
# tftp://jurai/tclware/au/en/ch_lang_sel1.au
```

Debit Card Prerequisite Tasks

Before you can configure your access server (Cisco AS5300 universal access server, Cisco 3600 series routers, or other supported voice platform) with the Debit Card for Packet Telephony feature, perform the following tasks:

- Establish a working IP network. For more information about configuring IP, refer to the "IP Overview," and "IP Addressing and Services" chapters in the *Cisco IOS IP Routing Configuration Guide*.
- Configure VoIP for the service provider environment. In addition to the basic configuration tasks, such as configuring dial peers and voice ports, you must configure specific devices in your network to act as gateways and gatekeepers. For more information about configuring gatekeepers, refer to the "Configuring H.323 Gatekeepers" chapter. For more information about configuring gateways, see the "Configuring H.323 Gateways" chapter.
- Configure a TFTP server to perform storage and retrieval of the audio files.
- Download the appropriate classic or IVR TCL script from the Cisco.com and store the scripts and audio files on the TFTP server configured to interact with your gateway access server.
- Make sure that your audio files are in the proper format. The IVR prompts require audio file (.au) format of 8-bit, u-law, and 8 Khz encoding. If you want to encode your own audio files, we recommend that you use one of these two audio tools (or a similar tool of equivalent quality):
  - Cool Edit, manufactured by Syntrillium Software Corporation
  - AudioTool, manufactured by Sun Microsystems
- Ensure that your access platform has a minimum of 16 MB Flash and 64 MB of DRAM.
- Install and configure the appropriate RADIUS security server in your network. The version of RADIUS that you are using must be able to support IETF-Supported VSAs, which are implemented by using IETF RADIUS attribute 26.

Debit Card for Packet Telephony Configuration Tasks List

Configure the Debit Card for Packet Telephony feature the same way you configure IVR because the feature uses the IVR infrastructure. To configure this feature, you need to perform the following tasks:

- Create an application that will interact with the appropriate classic or TCL script. See the "Configuring IVR Applications" chapter.
- Define and pass the defined parameter values to the application. These values include the language of the audio file, the location of the audio file, the designated operator telephone number of the service provider (redirect number), the number of characters in the PIN, the number of characters in the user identification number (UID), the number of times a caller is permitted to reenter the PIN (retry count), and the number of seconds of warning a user receives before the allowed calling time runs out.
- Associate the application to the incoming POTS dial peer.
- Define the appropriate method lists using AAA so that you identify RADIUS as the security protocol performing accounting.

To configure Debit Card for Packet Telephony, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call application voice application-name location`	Defines the name to be used for your application and indicates the location (URL) of the appropriate IVR script to be used with this application.
Step 2	`Router(config)# call application voice application-name language language`	Defines the language of the audio file for the designated application and passes that information to the application.
Step 3	`Router(config)# call application voice application-name redirect-number number`	Defines the telephone number to which a call will be redirected—for example, the operator telephone number of the service provider—for the designated application and passes that information to the application.
Step 4	`Router(config)# call application voice application-name pin-length number`	Defines the number of characters in the PIN for the designated application and passes that information to the application.
Step 5	`Router(config)# call application voice application-name retry-count number`	Defines the number of times a caller is permitted to reenter the PIN for the designated application and passes that information to the application.
Step 6	`Router(config)# call application voice application-name uid-length number`	Defines the number of characters in the UID for the designated application and passes that information to the application.
Step 7	`Router(config)# call application voice application-name warning-time seconds`	Defines the number of seconds of warning a user receives before the allowed calling time runs out for the designated application and passes that information to the application.
Step 8	`Router(config)# call application voice application-name set-location language category location`	Defines the location, language, and category of the audio files for the designated application and passes that information to the application.

	Command	Purpose
Step 9	`Router(config)# aaa new-model`	Enables AAA security and accounting services.

Step 10	`Router(config)# gw-accounting h323`
	or
	`Router(config)# gw-accounting syslog`
	or
	`Router(config)# gw-accounting vsa`
		Configures gateway-specific H.323 accounting. The **h323** keyword configures standard H.323 accounting using standard IETF RADIUS attributes. The **syslog** keyword configures the system logging facility to output accounting information in the form of a system log message. The **vsa** keyword configures the VSA method of applying H.323 gateway-specific accounting.

Step 11	`Router(config)# aaa authentication login h323 radius`	Defines a method list called h323 where RADIUS is defined as the only method of login authentication.
Step 12	`Router(config)# aaa accounting connection h323 start-stop radius`	Defines a method list called h323 where RADIUS is used to perform connection accounting, providing start-stop records.
Step 13	`Router(config)# radius-server host` *ip-address* `auth-port` *number* `acct-port` *number*	Identifies the RADIUS server and the ports that will be used for authentication and accounting services.
Step 14	`Router(config)# radius-server key` *key*	Specifies the password used between the gateway and the RADIUS server.
Step 15	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode to configure the incoming POTS dial peer. **Note** The *number* value of the **dial-peer voice pots** command is a tag that uniquely identifies the dial peer.
Step 16	`Router(config-dial-peer)# application` *application-name*	Associates the IVR application with the incoming POTS dial peer.
Step 17	`Router(config-dial-peer)# destination-pattern [+]`*string* `T`	Defines the telephone number associated with this dial peer.
Step 18	`Router(config-dial-peer)# port` *port-number*	Defines the voice port associated with this dial peer.

Note Because Cisco security authenticates based on account number, RADIUS is required for the redialer fax application.

Note RADIUS is turned on globally but is only used for services if it is so programmed.

Verifying the Debit Card Configuration

You can verify Debit Card for Packet Telephony configuration by performing the following tasks:

- To verify that the newly created application is listed, use the **show call application voice summary** command.
- To verify that the application associated with the dial peer is correct, use the **show dial-peer voice** command.

Debit Card Feature Configuration Example

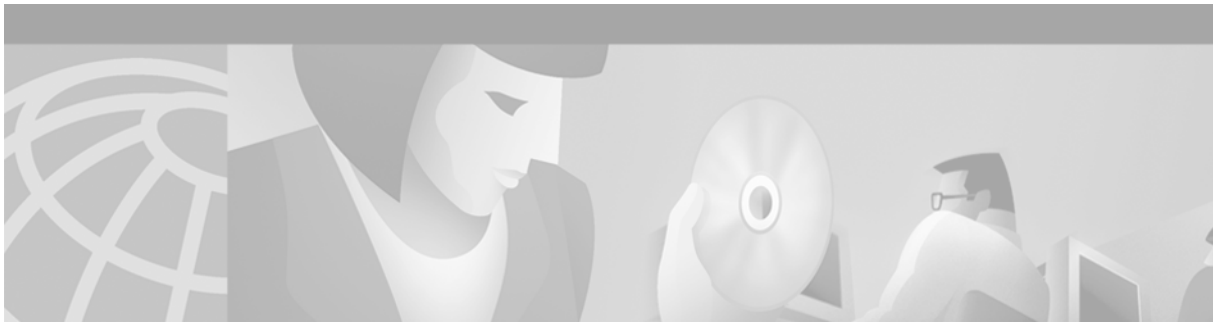
The following example displays the configuration for the debit card feature; this output was creating by using the **show running-config** command:

```
Router # show running-config
Building configuration...

Current configuration:
!
version 12.2
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service internal
!
hostname Router name
!
no logging buffered

! AAA configuration
!-------------------------------------
aaa new-model
aaa authentication login h323 group radius
aaa authorization exec h323 group radius
aaa accounting connection h323 start-stop group radius
!-----------------------------------
!
enable secret 5 $1$rLpz$DpgRh8qfaDqCPteN4/KXD0
enable password xxx
!
username Router password 0 xyxyxy
username s
!
resource-pool disable
!
ip subnet-zero
no ip domain-lookup

!  TFTP address configuration
!---------------------------------
ip host keyer 223.255.254.254

! prepaid application creation
!-----------------------------------
call application voice prepaid tftp://keyer/debitcard.tcl

! passing parameters to prepaid application
!---------------------------------
```

```
call application voice prepaid uid-len 4
call application voice prepaid language 1 en
call application voice prepaid language 2 ch
call application voice prepaid set-location en 0 tftp://keyer/

mta receive maximum-recipients 1024
!
dial-control-mib max-size 300
!
controller T1 0
 shutdown
 framing esf
 linecode b8zs
 cablelength short 133
!
controller T1 1
 shutdown
 framing esf
 linecode b8zs
 cablelength short 133
!
controller T1 2
 framing esf
 clock source line primary
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
!
controller T1 3
 framing esf
 clock source line secondary 1
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
!
!
voice-port 2:D
 timeouts call-disconnect 0
!
voice-port 3:D
 timeouts call-disconnect 0

! configuring voip gw accounting
!-------------------------------
gw-accounting h323 vsa

! associating application to dial-peer
!-------------------------------------
dial-peer voice 30001 pots
 application prepaid
 destination-pattern 300..
 port 2:D
 prefix 300
!
dial-peer voice 40001 pots
 destination-pattern 400..
 direct-inward-dial
 port 3:D
 prefix 400
!
dial-peer voice 50001 voip
 destination-pattern 500..
 session target ipv4:147.14.25.1
!
```

```
dial-peer voice 60001 voip
 destination-pattern 600..
 session target ipv4:147.14.25.1
!
process-max-time 200
!
interface Ethernet0
 description ip address 132.132.1.2 255.255.255.0
 ip address 1.13.103.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 no keepalive
 no cdp enable
!
interface Serial2:23
 description D-Channel - To Abacus
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface Serial3:23
 description D-Channel - To Abacus
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 ip address 147.14.25.100 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 duplex full
 no cdp enable
 hold-queue 2048 in
!
interface Async1
 ip address 2.2.2.1 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 shutdown
 async mode dedicated
 ppp authentication chap
 hold-queue 10 in
!
interface Group-Async1
 physical-layer async
 ip unnumbered Serial2:22
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 dialer in-band
 dialer idle-timeout 200000
 async default routing
 async mode interactive
```

```
 no peer default ip address
 no fair-queue
 no cdp enable
 hold-queue 10 in
!
router igrp 200
 network 1.0.0.0
 network 133.133.0.0
!
router igrp 300
 network 132.132.0.0
 network 133.133.0.0
 network 147.14.0.0
!
no ip http server
no ip classless
!
ip route 1.13.80.100 255.255.255.255 1.13.0.1
ip route 223.255.254.254 255.255.255.255 Ethernet0
!
!
logging history size 500

! configuring radius parameters
!---------------------------------------
radius-server host 1.13.80.100 auth-port 1812 acct-port 1813
radius-server key cisco
radius-server vsa send accounting
radius-server vsa send authentication

!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 exec-timeout 0 0
 password lab

! configuring the NTP
!------------------------------
ntp master 15

!
end
```



Configuring Settlement Applications

The Cisco Settlement for Packet Telephony feature equips Cisco conferencing infrastructure products to use third-party settlement systems on multiple protocols. The Settlement for Packet Telephony feature allows Internet telephony service providers to do the following:

- Act as clearinghouses to validate and reconcile billing information from different sources and occurrences so that the service providers can produce separate billing statements for each call party
- Provide functions such as call routing, authentication, reconciliation, and the settlement solution in multiple currencies.
- Enable Cisco access platforms to provide Open Settlement Protocol (OSP) for service providers
- Work with the existing AAA feature to provide security and accounting services
- Specify a list of patterns that can be matched with a user account to see if that user is roaming
- Limit calls to authorized users and prevents unauthorized usage of limited telephony resources
- Allow users to initiate a Voice over Internet Protocol (VoIP) telephone connection from a web server page

Cisco provides a set of enabling technologies for Cisco IOS products to interface with third-party settlement systems.

The Settlement for Packet Telephony feature complies with the European Telecommunication Standards Institute (ETSI) Technical Specification (TS) 101 321.

This chapter contains the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information mentioned in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Settlement for Packet Telephony Overview

When you make a telephone call, the cost charged can be divided among various carriers involved in the completion of the call. *Settlement* is the method used to divide the cost among the carriers. Traditionally, settlement agreements have been arranged between pairs of carriers. With the advance of voice and video conferencing over IP, pairwise settlement agreements have become cumbersome. A number of companies have entered the market offering settlement on a subscription basis. As a result, the settlement process has become a more manageable, many-to-one system, with a set of public interfaces implemented by service providers.

The Cisco gateway-based settlement protocol interacts between carriers to create a single authentication at initialization. The authentication is the basis for the establishment of a secure communication channel between the settlement system and the infrastructure component. This channel then allows the following three types of transactions to be handled:

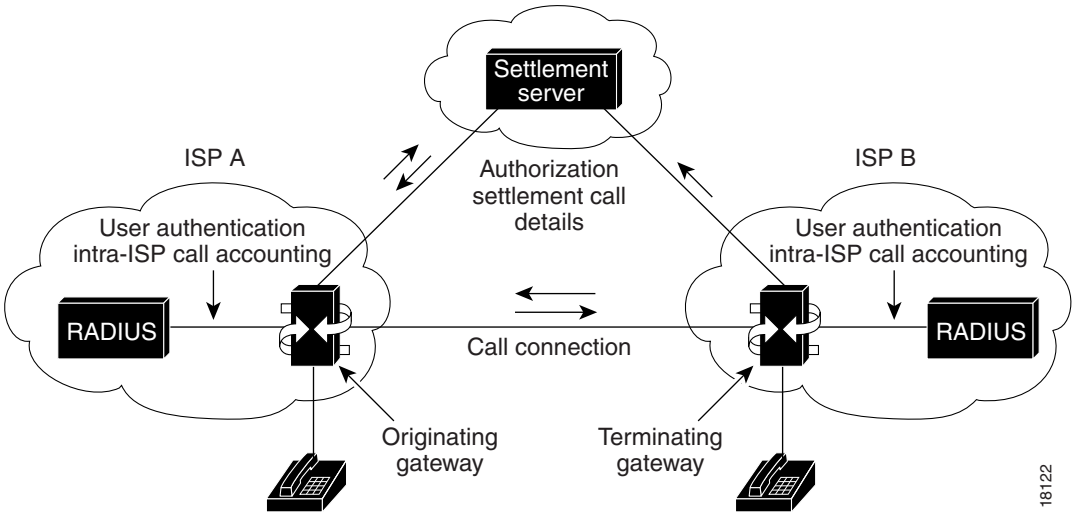
- Call routing. The settlement system can either accept a gateway endpoint from the requestor or assign one for the requestor.
- Call authorization. Based on the terminating endpoint address, the settlement system determines whether the requesting gateway is permitted to originate calls for the terminating gateway. If the call is authorized, the settlement system generates a token that allows the terminating gateway to accept the call.
- Call detail reporting. Each endpoint in a call leg reports when the call stops, along with the usual call details. The settlement system reconciles the various reports of the calling and called parties and generates billing information. Call details are reported on a call-by-call basis.

Figure 116 shows a typical gateway–based settlement network topology. A voice or fax call is originated and routed through the gateway (a Cisco AS5300 Universal Access Server or a Cisco 2600 or Cisco 3600 series router) to a database server (RADIUS or TACACS+) for user authentication and intra-internet service provider (ISP) call accounting. Using tool command language (TCL) interactive voice response (IVR) scripts to gather and manipulate the caller's data, the gateway forwards the call to the settlement server, which authorizes the call and adds settlement details in a token. The call, now carrying its unique settlement token, passes through the originating gateway to the terminating gateway. The terminating gateway uses TCL IVR to validate the settlement token and forwards the call to the receiving telephone or fax machine.

Note For a complete description of the IVR feature, see the chapter "Configuring TCL IVR Applications."

When the call is completed, both the terminating and originating gateways communicate the call details to the settlement server. The settlement server then reconciles the information it receives about the call from both gateways.

Figure 116 Gateway-Based Settlement



Settlement (OSP) Enhancements

Since the introduction of Settlements for Packet Voice, the Settlements for Packet Voice protocol has also undergone several feature enhancements. These enhancements are documented in the following sections:

- Roaming, page 537
- Public Key Infrastructure Multiple Roots, page 539
- User-Network Interface OSP, page 540

Roaming

Support for the settlement functions required for roaming callers has been added. A caller is roaming when dialing into a gateway that is not the home gateway. A home gateway belongs to the user's service provider. Usually, the subscriber is billed with additional charges for roaming calls. The settlement server and the service provider need to know when a caller is roaming in order to create accurate billing statements.

A roaming caller must be authenticated before a call can go through a gateway. Both authentication, authorization, and accounting (AAA) and the settlement server can authenticate a roaming user. If AAA fails to authenticate a roaming caller, the roaming call must be routed to a settlement server. If the settlement server cannot authenticate the caller, the call is terminated.

You can use the following methods to configure the roaming feature on the gateway:

- Setting the roaming patterns to determine if a caller (by user identification) is roaming
- Setting the roaming capability in the settlement provider
- Setting the roaming capability in the dial peer
- Forcing a call to be routed via a settlement server in a dial peer

User Identification

The gateway can specify a list of patterns to be matched with a user account number (user identification) to see if that user is roaming. The user enters an account number and personal identification number (PIN) as part of the interaction with the TCL IVR prompts.

The roaming patterns are configured by using the **settlement roam-pattern** command in global configuration mode.

For additional information about the IVR or AAA, refer to the following Cisco IOS documents:

- *Cisco Interactive Voice Response*
- *Service Provider Features for Voice over IP*

Settlement Provider

Some settlement providers want to know if a user is roaming so they can apply the appropriate charge to a user account. Other settlement providers do not distinguish between local and roaming users.

A settlement provider can use the **roam** command in the settlement configuration mode to track roaming users. If a user is roaming and the settlement provider is tracking roaming, the gateway sends the user account number and PIN to the settlement server so that the user can be properly authenticated.

Dial Peer

A gateway can dictate if a particular outbound dial peer can terminate roaming calls and only permit local calls with the **no roam** command. The default of the dial peer is no roaming support. The gateway allows a roaming call to go through only if both the dial peer associated with that call and the settlement provider support roaming. In other words, a call fails if the dial peer has roaming enabled but the settlement provider does not, and vice versa. Therefore, the roaming feature must be explicitly enabled in the dial peer.

Dial Peer Settlement Option

The **settle-call** keyword forces the call to go through a settlement server regardless of the session target type. If the session target type is ipv4, dns, or RAS, the gateway resolves the terminating gateway address and asks the settlement server to authorize that terminating gateway.

The restrictions and behaviors associated with use of the **settle-call** keyword with outbound dial peers are described in the “Restrictions” section later in this chapter.

Public Key Infrastructure Multiple Roots

The public key infrastructure (PKI) multiple roots allows a settlement server to use one certificate for a Secure Socket Layer (SSL) handshake and a different certificate for token signing. Cisco devices can share public keys using digital certificates.

Digital certificates are normally issued by trusted third parties, which are called certificate authorities (CAs). Every router that uses digital certificates should enroll its public key with the CA server. Typically during enrollment, the certificate administrator (a person) will manually verify that the requesting router is authentic and grant the certificate; some CA servers can authenticate the routers automatically.

A certificate has many fields, including a serial number, a fingerprint, and an expiry date. A certificate can be revoked before its expiry date because of key compromise or other security reasons. The CA server maintains a list of revoked certificates, which is called a certificate revocation list (CRL). Routers can be configured not to accept a peer certificate that has been revoked. A router downloads a CRL from the CA server for this purpose.

Cisco routers use a proprietary Certificate Enrollment Protocol (CEP) to communicate with the CA server. The CA server should understand CEP.

The PKI Multiple Roots feature is based on the Cisco security and PKI technology. For in-depth information about security, refer to the *Cisco IOS Security Configuration Guide*.

Different commands are used for the following purposes (as follows):

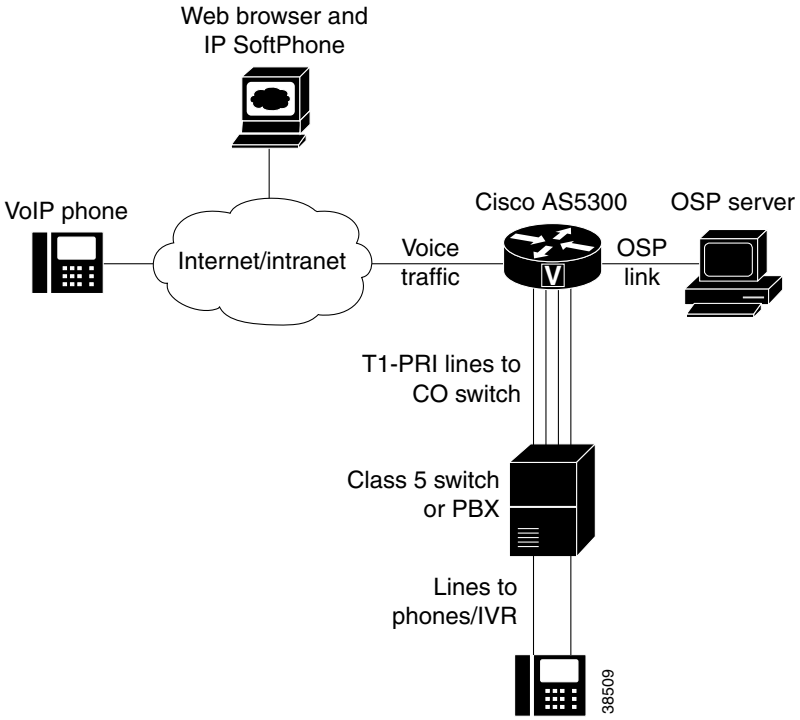
- For SSL handshake with the settlement server, the gateway uses the certificate obtained through the **crypto ca authenticate** command.
- For token verification, the gateway can use one of the root certificates configured with the **crypto ca trusted-root** *identity* command.

Note To specify which root certificate is used for token validation, use the command **token-root-name** in the settlement configuration mode.

User-Network Interface OSP

The User-Network Interface (UNI)-OSP feature, illustrated in Figure 117, allows a single Cisco AS5300 gateway to use OSP to authenticate VoIP calls to the PSTN.

Figure 117 Authenticating VoIP Calls with the Cisco AS5300 Gateway



To implement VoIP authentication, the gateway must be connected to the Internet or intranet and to an OSP server. The OSP server, which can be any properly configured Windows NT or UNIX server, communicates over a Computer Telephony Interface (CTI) link to a Class 5 switch or PBX. Use the settlement command **type** and include the *uni-osp* argument to configure the UNI-OSP feature.

When a VoIP device sends an H.323 setup message to the gateway, the destination number (DNIS) of the call is matched to a POTS dial peer configured on the voice gateway. The voice gateway then sends an OSP authorization request, containing the call ID (a unique16-bit number), and a calling and called number (E.164 ANI/DNIS).

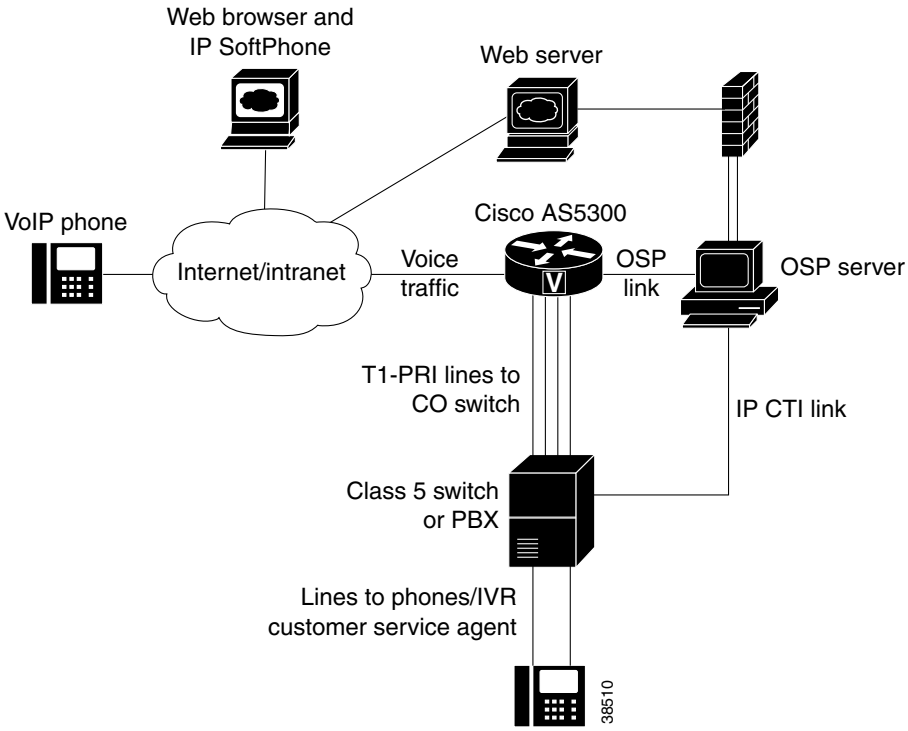
The OSP server sends an authorization response back to the voice gateway, which then initiates a call to a PBX or Class 5 switch over one of its T1-PRI spans. When the voice gateway detects that a call has ended, it transmits usage information to the OSP server, informing it that the call has terminated.

Note that the “src-info” field of the OSP authorization request contains the IP address of the caller’s PC, with all “dot” characters removed and each three-digit segment right justified. For example, the automatic number identification (ANI) field for a call originating from IP address 172.16.1.20 would appear as “172069221002.”

Click-to-Talk Functionality

As illustrated in Figure 118, the UNI-OSP feature can be used when implementing a "click-to-talk" function on web server pages.

Figure 118 Implementing Click-to-Talk with the Cisco 5300 Gateway



When using the click-to-talk function, a customer selects a link on a web page indicating that a request to talk to a customer service or technical support representative. The web server then launches a preinstalled SoftPhone on the web browser machine through a browser plug-in. The web server supplies the PC SoftPhone application with the telephone destination number (DNIS) of the appropriate agent, and the route point, queue, and IP address of the voice gateway.

When the SoftPhone sends an H.323 setup message to the voice gateway, the destination number (DNIS) is matched to a POTS dial peer configured on the voice gateway. An authorization request is then sent over the OSP link to the OSP server, containing the call ID (a unique16 bit number) and a calling and called number (E.164 ANI/DNIS). Because the originating device is a PC SoftPhone, the ANI field contains the IP address of the PC.

The OSP server compares the IP address received in the ANI field with those customers who have pressed the click-to-talk link. The OSP server sends an authorization response to the voice gateway, containing the E.164 number of the appropriate agent, based on the web page from which click-to-talk was initiated.

When the voice gateway initiates a call to a PBX or Class 5 switch, the arriving call causes a setup indication to appear on the switch or PBX. The CTI link between the PBX or switch and the OSP server informs the OSP server of the incoming call and includes information such as the DNIS, ANI (IP address of the caller's PC), and the incoming trunk line. The OSP server then has sufficient information to route the call to the appropriate customer service or technical support agent queue.

Settlement for Packet Telephony Prerequisite Tasks

Before you can configure your access server platform (Cisco AS5300 universal access server, Cisco 3600 series routers, or other supported voice platform) with the Settlement for Packet Telephony feature, you must perform the following tasks:

- Ensure that your access platform has a minimum of 16 MB Flash memory and 64 MB DRAM.
- In Cisco IOS Release 12.0(4)XH or later releases, both the originating and terminating gateways must be using the TCL IVR scripts to perform settlement successfully. If a terminating gateway that is not configured with a TCL script receives settlement calls, it will not recognize the tokens added to those calls by the settlement server; therefore, those calls will pass through without being audited or charged.
- Ensure that the correct version of VCWare is downloaded.
- Before configuring the settlement feature, you must have configured the PKI for secured communication between the access platform (or router) and the settlement server. For detailed information about certificates and secure devices, refer to the Cisco IOS Release 12.0 document titled *Certification Authority Interoperability.*

Restrictions

- The Settlements for Packet Voice, Phase 2, feature requires Cisco IOS Release 12.1(1)T and a compatible version of VCWare.

Note The Cisco AS5800 universal access server uses portware, not VCWare, with its modems.

- The Settlements for Packet Voice, Phase 2, feature set cannot be enabled on dial peers that use remote access server (RAS) as the session target.
- The software that includes the Settlements for Packet Voice, Phase 2, feature set is offered only in crypto images and therefore is under export controls.

Settlement for Packet Telephony Configuration Task List

To configure settlement for packet telephony, perform the following tasks:

- Configuring the Public Key Infrastructure, page 543
- Configuring the Originating Gateway, page 544
- Configuring the Terminating Gateway, page 548
- Configuring Settlement with Roaming, page 551
- Configuring Settlement with PKI Multiple Roots, page 552
- Configuring Settlement with Suggested Route, page 553

Configuring the Public Key Infrastructure

Note Ensure that you have secure communication between the access platform or router and the settlement server.

To configure the PKI, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# no crypto ca identity name`	Clears the old CA identity if a previous one exists.
Step 2	`Router(config)# crypto key zeroize rsa`	Clears the existing RSA key.
Step 3	`Router(config)# hostname router-name`	Configures the host name of the router if this has not been done already.
Step 4	`Router(config)# ip domain-name domain-name`	Configures the IP domain name of the router.
Step 5	`Router(config)# ip host CA-hostname CA-ipaddress`	Enters the CA host name and IP address.
Step 6	`Router(config)# crypto ca identity name`	Declares a CA name and enters CA-identity configuration mode. For example, the *name* argument could be fieldlabs.cisco.com.
Step 7	`Router(ca-identity)# enrollment url url`	The /cgi-bin/pkiclient.exe file is the default Common Gateway Interface (CGI) script that Cisco IOS software assumes. The script path should be given in the URL if it is different from the default. **Note** The URL should have the format http://CA-hostname where CA-hostname is previously configured in Step 5.
Step 8	`Router(ca-identity)# enrollment retry count number`	(Optional) Specifies how many times the router will poll the CA server for the certificate status when the certificate requests are pending. **Note** The router sends the certificate request only once. Then it periodically polls the CA server until the certificate is granted or denied, or until the retry count exceeds the retry count configured.
Step 9	`Router(ca-identity)# enrollment retry period minutes`	(Optional) Specifies the interval between subsequent polls. Default = 1 minute. **Note** The retry period contains two subsequent polls for certificate status. The router does not send another certificate request. It merely polls for the status as long as the CA server returns the certificate status as pending, or until the retry count is reached. **Note** After specifying a certificate, the router waits to receive a certificate from the CA. If the router does not receive a certificate within a period of time (the retry period), the router sends another certificate request.
Step 10	`Router(ca-identity)# exit`	Exits CA-identity configuration mode.

	Command	Purposes
Step 11	`Router(config)# crypto ca authenticate name`	Obtains the CA's public key. Use the same *name* that you used when declaring the CA with the **crypto ca identity** command.
Step 12	`Router(config)# crypto key generate rsa`	Generates the RSA key pair.
Step 13	`Router(config)# crypto ca enroll name`	Obtains the router certificate for all your RSA key pairs. **Note** This command requires you to create a challenge password that is not saved with the configuration. This password is required in order to obtain a new certificate if your certificate is revoked, so remember this password. **Note** If your router reboots after you issue the **crypto ca enroll** command but before you receive the certificate, you must reissue the command.

Configuring the Originating Gateway

To configure the originating gateway, perform the following tasks:

- Configuring the Settlement Provider, page 544
- Configuring the Inbound POTS Dial Peer, page 545
- Configuring the Outbound VoIP Dial Peer, page 547

Configuring the Settlement Provider

To configure the settlement provider to authorize calls, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# settlement provider`	Enters settlement configuration mode and configures the settlement provider number.
Step 2	`Router(config-settlement)# type osp`	Configures the settlement provider type.
Step 3	`Router(config-settlement)# url url-address`	Enters the settlement provider URL for the ISP that is hosting the settlement server. **Note** This step can be repeated if the settlement provider has more than one service point.
Step 4	`Router(config-settlement)# response-timeout number`	Configures the maximum time, in seconds, to wait for a response from a server. The default response timeout is 1 second.
Step 5	`Router(config-settlement)# no shutdown`	Activates the settlement provider.

Note If you are configuring a TransNexus server, first enter the **url-address** command, and then enter the **customer-id** and **device-id** commands.

Configuring the Inbound POTS Dial Peer

Note In Step 2 of the following procedures, do not use the default session application. The default session application does not support settlement. Calls handled by the default session application are not routed to a settlement server. Settlement tokens are not validated in the default session application.

To configure the inbound POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode to configure a POTS dial peer. The *number* argument uniquely identifies the dial peer.
Step 2	`Router(config-dial-peer)# application` *application-name*	Configures the application attribute and identifies the desired TCL script using the *application-name* argument. **Note** The *application-name* must be the name of the TCL IVR script. If the application attribute is not configured, or if the POTS dial peer is not created, the default session application will process the call.

	Command	Purposes
Step 3	`Router(config-dial-peer)#` **`destination-pattern`** [**+**]*`string`*[**T**]	Configures the destination pattern of the dial peer. Enter the number or pattern of the outbound called number. The *string* argument is a series of digits that specify an E.164 or private dialing plan telephone number. Valid entries are the digits 0–9 and the letters A–D. The following special characters can be entered in the string: • Plus sign (**+**)—(Optional) Indicates an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Specifies the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – Asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. On the Cisco 3600 only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,) inserts a pause between digits. – Period (.) matches any entered digit (this character is used as a wildcard). On the Cisco 3600, the period cannot be used as a leading character in a string (for example, .650). • **T**—(Optional) Indicates that the destination-pattern value is a variable length dial-string.
Step 4	`Router(config-dial-peer)#` **`port`** *`port-number`*	Associates this voice-telephony dial peer with a specific voice port.

Configuring the Outbound VoIP Dial Peer

To configure the outbound VoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters dial-peer configuration mode to configure the outbound VoIP dial peer. The *number* argument uniquely identifies the dial peer.
Step 2	`Router(config-dial-peer)# destination-pattern [+]`*string*`[T]`	Configures the destination pattern of the dial peer. Enter the number or pattern of the outbound called number. The *string* is a series of digits that specify an E.164 or private dialing plan telephone number. Valid entries are the digits 0–9 and the letters A–D. The following special characters can be entered in the string: • Plus sign (**+**)—(Optional) Indicates an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Specifies the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – Asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. On the Cisco 3600 only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,) inserts a pause between digits. – Period (.) matches any entered digit (this character is used as a wildcard). On the Cisco 3600, the period cannot be used as a leading character in a string (for example, .650). • **T**—(Optional) Indicates that the destination-pattern value is a variable length dial-string.
Step 3	`Router(config-dial-peer)# session target settlement`	Configures settlement as the target to resolve the terminating gateway address. **Note** The *provider-number* argument should match one of the number values previously configured in Step 1.

Note The originating gateway system clock must synchronize with the settlement server clock. Use the **clock** or **ntp** command to set the router clock.

Configuring the Terminating Gateway

Caution If the terminating gateway is not configured to use TCL IVR application scripts, the settlement tokens are bypassed, calls can get through, and settlement calls will not be audited; therefore, you will not be notified that the calls are not going through the billing service.

To configure the terminating gateway, perform the following tasks:

- Configuring the Settlement Provider, page 548
- Configuring the Inbound VoIP Dial Peer, page 549
- Configuring the Outbound POTS Dial Peer, page 550

Configuring the Settlement Provider

To configure the settlement provider, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# settlement` *provider-number*	Enters settlement configuration mode and configures the settlement provider number.
Step 2	`Router(config-settlement)# type osp`	Configures the settlement provider type.
Step 3	`Router(config-settlement)# url` *url-address*	Enters the settlement provider URL for the ISP hosting the settlement server. **Note** This step can be repeated if the settlement provider has more than one service point.
Step 4	`Router(config-settlement)# response-timeout` *number*	Configures the maximum time, in seconds, to wait for a response from a server. The default response timeout is 1 second.
Step 5	`Router(config-settlement)# no shutdown`	Activates the settlement provider.

Note If you are configuring a TransNexus server, enter the **url** command, and then enter the **customer-id** and **device-id** commands.

Configuring the Inbound VoIP Dial Peer

Note The default session application does not support settlement. Calls handled by the default session application are not routed to a settlement server. Settlement tokens are not validated in the default session application.

To configure the inbound VoIP dial peer, perform the following tasks beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters the dial-peer configuration mode to configure the inbound VoIP dial peer. The *number* argument uniquely identifies the dial peer.
Step 2	`Router(config-dial-peer)# application` *application-name*	Configures the application attribute and identifies the desired TCL script using the *application-name* argument.
Step 3	`Router(config-dial-peer)# incoming called-number` *string*	Specifies the telephone number of the voice port associated with this dial peer. String characters include wildcards to create the number or pattern.
Step 4	`Router(config-dial-peer)# session target settlement` *provider-number*	Identifies settlement as the session target to resolve the terminating gateway address. **Note** The *provider-number* value should match one of the number values previously configured in Step 1 of the section "Configuring the Settlement Provider".

Configuring the Outbound POTS Dial Peer

To configure the outbound POTS dial peer, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode to configure the outbound POTS dial peer. The *number* argument uniquely identifies the dial peer.
Step 2	`Router(config-dial-peer)# destination-pattern [+]`*string*`[T]`	Configures the destination pattern of the dial peer pattern. Use the called number. The *string* is a series of digits that specify an E.164 or private dialing plan telephone number. Valid entries are the digits 0–9 and the letters A–D. The following special characters can be entered in the string: • Plus sign (**+**)—(Optional) Indicates an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810 multiservice concentrator. • *string*—Specifies the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – Asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. On the Cisco 3600 only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,) inserts a pause between digits. – Period (.) matches any entered digit (this character is used as a wildcard). On the Cisco 3600, the period cannot be used as a leading character in a string (for example, .650). • **T**—(Optional) Indicates that the destination-pattern value is a variable length dial-string.
Step 3	`Router(config-dial-peer)# port` *port-number*	Associates the voice-telephony dial peer with a specific voice port. Activates the voice port associated with this dial peer.

Note The terminating gateway system clock must synchronize with the settlement server clock. Use the **clock** or **ntp** command to set the router clock.

Verifying Settlement Configuration

Use the **show running-config** command to verify your configuration.

Configuring Settlement with Roaming

To configure settlement with the roaming capability, perform the configuration tasks described in the following sections:

Configuring the Roaming Patterns on the Originating Gateway

To configure the roaming patterns on the originating gateway, use the following commands beginning in global configuration mode:

Command	Purposes
`Router(config)# settlement roam-pattern` *pattern*	Defines the pattern for roaming account numbers. Enter multiple instances of this command to specify multiple patterns.

Enabling the Roaming Feature for the Settlement Provider

To enable the roaming feature for the settlement provider, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# settlement` *provider-number*	Enters settlement configuration mode and configures the settlement provider number.
Step 2	`Router(config-settlement)# roaming`	Enables the roaming capability on this provider.

Enabling the Roaming Feature in the Outbound Dial Peer

To enable the roaming feature in the outbound dial peer, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# dial-peer voice` *number* `voip`	Enters dial-peer configuration mode to configure a VoIP dial peer.
Step 2	`Router(config-dial-peer)# roaming`	Enables roaming on this dial peer.

Configuring Settlement with PKI Multiple Roots

To configure the PKI multiple roots capability, perform the configuration tasks described in the following sections:

Configuring a Settlement Server with PKI Multiple Roots on the Originating Gateway

To configure a settlement server with PKI Multiple Roots on the Originating Gateway, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# settlement provider-number`	Enters settlement configuration mode for a specific provider.
Step 2	`Router(config-settlement)# url url-address`	Enters the URL to the service point that uses two different certificates for SSL and token.

Configuring the Root Certificate for Token Validation on the Terminating Gateway

To configure the root certificate for token validation on the Terminating Gateway, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# crypto ca trusted-root identity`	Configures the root certificate that the server uses to sign the settlement tokens.
Step 2	`Router(ca-root)# root tftp tftp-ipadress root-ca-file`	Specifies where to obtain the root certificate file.
Step 3	`Router(ca-root)# crypto ca authenticate identify-name`	Starts downloading the root certificate file from the server.

Defining the Token Validation on the Terminating Gateway

To define the token validation on the Terminating Gateway, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# settlement provider-number`	Enters settlement configuration mode for a specific provider.
Step 2	`Router(config-settlement)# token-root-name name`	Specifies which root certificate the gateway uses to validate the token. The name must match the name of the certificate configured using either the **crypto ca identity** *name* or the **crypto ca trusted-root** *identity* commands.

Configuring Settlement with Suggested Route

The **session target** command in the dial peer dictates how the gateway resolves the terminating address to complete the call. Besides settlement, the gateway could use the **ipv4** or **dns** options if it knows the exact address of the Terminating Gateway, or it could use the **ras** option to consult a gatekeeper.

To force a call to be authorized by a settlement server, use the following commands beginning in global configuration mode:

	Command	Purposes
Step 1	`Router(config)# dial-peer voice number voip`	Enters dial-peer configuration mode to configure a VoIP dial peer.
Step 2	`Router(config-dial-peer)# settle-call [provider-number]`	Forces a call to be authorized with a settlement server that uses the address resolution method specified in the **session target** *type* command.

Table 42 shows how settlement is enabled on a dial peer based on various combinations of the **session target** and **settle-call** commands.

Table 42 The settle-call and session target Commands

Command	session target IP DNS	session target settlement	session target RAS
settle-call	Settlement processing will occur; see Table 43.	Settlement processing will occur; see Table 43.	Illegal (legal once cc_ResolveAddress function is implemented).
no settle-call	Settlement processing will *not* occur; see Table 43.	Settlement processing will occur; see Table 43.	Settlement processing will *not* occur; see Table 43.

Note If the **session target settlement** and **settle-call** keywords are used, the keywords must be the same or an error is generated. If one Cisco IOS command specifies *one keyword* and the other does not, the specified *keyword* becomes the only clearinghouse used. If neither specifies a *keyword*, all clearinghouses can be searched.

Table 43 shows the results of using the **session target settlement** command.

Table 43 Results of the session target settlement Command

User Status	The settle Command	The no settle Command
User is authenticated and local.	• Authorizes call • Routes call • Generates settlement CDR[1]	• Authorizes call • Routes call • Generates settlement CDR
User is authenticated and roaming.	• Authenticates roaming user • Authorizes call • Routes call • Generates settlement CDR	• Authenticates roaming user • Authorizes call • Routes call • Generates settlement CDR
User is roaming but not yet authenticated.	• Authenticates roaming user • Authorizes call • Routes call • Generates settlement CDR	• Authenticates roaming user • Authorizes call • Routes call • Generates settlement CDR

1. CDR = call detail record.

Table 44 shows the results of using the **session target ipv4** or **session target dns** command.

Table 44 Results of the session target IPV4 and session target DNS Commands

User Status	The settle Command	The no settle Command
User is authenticated and local.	• Authorizes call • Provides IP address in a "DestinationAlternate" field and fails call if settlement returns something different • Generates settlement CDR	Performs no settlement operations.
User is authenticated and roaming.	• Authorizes call • Provides IP address in a "DestinationAlternate" field and fails call if settlement returns something different • Generates settlement CDR	Performs no settlement operations. This dial peer is configured so that the administration can use roaming-enabled AAA but not settlement.
User is roaming but not yet authenticated.	• Authenticates call **Note** Authentication failure is possible and implies that the "place call" TCL verb must return a code that allows the script to loop back to recollect account information. • Authorizes call • Provides IP address in a "DestinationAlternate" field and fails call if settlement returns something different • Generates settlement CDR	Fails the call (the user is not authenticated and there is no facility to do so using settlement).

Table 45 shows the result of using the **session target ras** command with no token.

Note Settlement and RAS session targets are *illegal* in Cisco IOS Release 12.0(4)XH. Table 45 applies to releases that allow RAS automatic repeat request (ARQ)/ Advanced Communications Function (ACF) to be performed prior to calling settlement.

The gateway needs a way to decide whether the gatekeeper has done settlement authorization. The gateway checks to see if the returned ACF contains a settlement token. Table 45 applies to the case where no token is returned.

Table 45 Results of the session target RAS Command with No Token

User Status	The settle Command	The no settle Command
User is authenticated and local.	• Authorizes call • Provides RAS signal address in "DestinationAlternate" field and fails call if settlement returns something different • Generates settlement CDR	Performs no settlement operations.
User is authenticated and roaming.	• Authenticates user • Authorizes call • Specifies "destinationSignalAddr" in "OSP DestinationAlternate" field and fails call if CH returns something different • Generates settlement CDR	Performs no settlement operations.
User is roaming but not yet authenticated.	• Authenticates user • Authorizes call • Specifies "destinationSignalAddr" in "OSP DestinationAlternate" field and fails call if CH returns something different • Generates settlement CDR	Fails the call (there is no way to authenticate the user).

Table 46 shows the result of using the **session target ras** command with token. In Table 46, the ACF returns a valid token, indicating that the call has already been authorized and routed by settlement.

Note The roaming scenarios require that the "ARQ sourceAlternative" field be formatted with the user credentials.

Table 46 Results of the session target RAS Command with Token

User Status	The settle Command	The no settle Command
User is authenticated and local.	• Generates settlement CDR only	Fails the call (implies that the dial peer was not configured to work with a settlement-enabled gatekeeper).
User is authenticated and roaming.	• Generates settlement CDR only	Fails the call (implies that the dial peer was not configured to work with a settlement-enabled gatekeeper).
User is roaming but not yet authenticated.	• Generates settlement CDR only	Fails the call (implies that the dial peer was not configured to work with a settlement-enabled gatekeeper).

Table 47 shows what happens when an incoming VoIP call is detected, depending on whether the setup message contains a token.

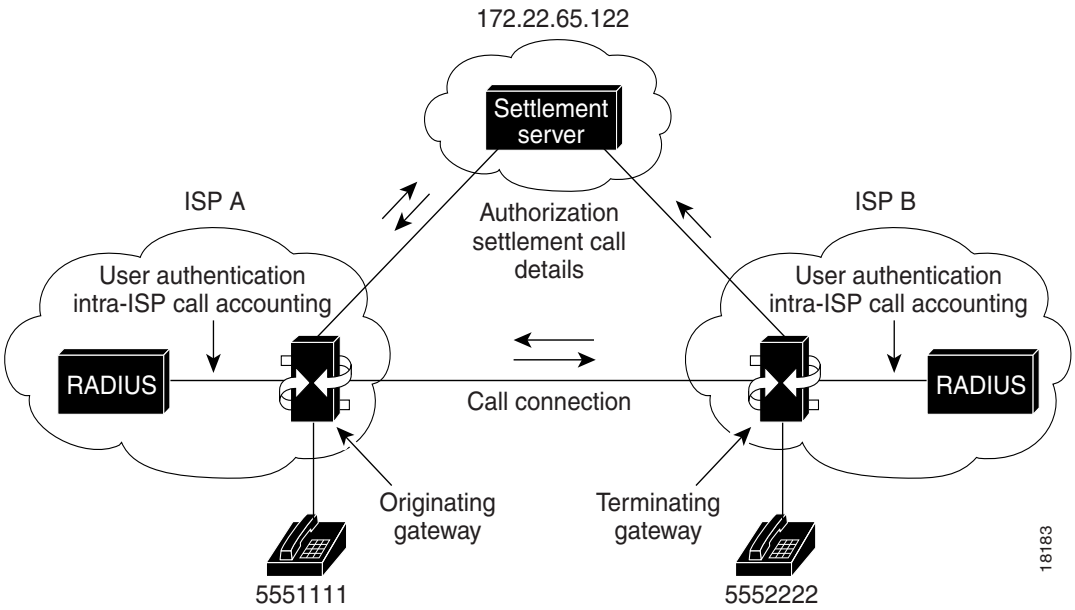
Table 47 Results of Receiving Inbound Calls

Token Status	The settle Command	The no settle Command
Settlement token is received in setup message.	• Validates token • Generates settlement CDR	Rejects the call (because the dial peer is not configured to do settlement, originated calls will not be settled).
No settlement token is received.	• Fails calls (to avoid fraudulent calls)	Accepts the call.

Settlement for Packet Telephony Configuration Examples

The examples that follow show settlement configurations for both the originating and terminating gateways. Figure 119 shows the topology for these configuration examples.

Figure 119 Example of Settlement Configurations for Originating and Terminating Gateways



Settlement on the Originating Gateway Example

The following example displays the configuration for the originating gateway by using the **show running-config** command:

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
service tcp-small-servers
!
hostname c3620-px15
!
ip subnet-zero
!
settlement 0
 type osp
 url http://xxx.xxx.
!
voice-port 1/0/0
 alerting audible
!
voice-port 1/0/1
 alerting audible
!
dial-peer voice 1 pots
 application session
 destination-pattern 5551111
 port 1/0/0
!
dial-peer voice 2 voip
 destination-pattern 5552222
 session target settlement:
!
interface Ethernet0/0
 ip address 172.22.65.131 255.255.255.224
 no ip directed-broadcast
 ip route-cache same-interface
 standby 1 priority 110
!
interface Serial0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet0/1
 no ip address
 no ip directed-broadcast
 shutdown
!
router eigrp 109
 network 172.22.0.0
!
router rip
 network 172.22.0.0
!
ip default-gateway 172.22.65.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.22.65.129
!
```

```
!
line con 0
 transport input none
line aux 0
line vty 0 4
 password
 login
!
end
```

Settlement on the Terminating Gateway Example

The following example displays the configuration for the terminating gateway resulting from the use of the **show running-config** command:

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
service tcp-small-servers
!
hostname 3620-px16
!
ip subnet-zero
ip domain-name cisco.com
ip name-server 198.92.30.32
!
settlement 0
 type osp
 url http://xxx.xxx.
!
voice-port 1/0/0
 alerting audible
!
voice-port 1/0/1
 alerting audible
!
dial-peer voice 1 pots
 destination-pattern 5552222
 port 1/0/0
!
dial-peer voice 2 voip
 application session
 incoming called-number 5552222
 session target settlement:0
!
interface Ethernet0/0
 ip address 172.22.65.143 255.255.255.224
 no ip directed-broadcast
 ip route-cache same-interface
!
interface Serial0/0
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Ethernet0/1
 no ip address
 no ip directed-broadcast
```

```
 shutdown
!
router eigrp 109
 network 172.22.0.0
!
router rip
 network 172.22.0.0
!
ip default-gateway 172.22.65.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.22.65.129
!
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 password
 login
!
end
```

Settlement with Roaming Example

The following output is the result of the settlement with roaming configuration:

```
!
version 12.2
service timestamps debug datetime
service timestamps log datetime
no service password-encryption
service internal
!
hostname as5300-05
!
enable secret 5 $1$lFSH$khsm3jB1lldHfXNlxqmaN1
enable password lab1
!
!resource-pool disable
!
!ip subnet-zero
ip host pkiserver 1.14.115.100
ip domain-name fieldlabs.cisco.com
ip name-server 172.16.1.4
!
isdn switch-type primary-5ess
isdn voice-call-failure 0
cns event-service server
mta receive maximum-recipients 1024
!
crypto cisco algorithm des
crypto cisco algorithm 40-bit-des
!
crypto ca identity transnexus
 enrollment retry count 100
 enrollment retry period 2
 enrollment url http://pkiserver:80
crypto ca certificate chain transnexus
 certificate ca 0171
  3082024C 308201B5 02020171 300D0609 2A864886 F70D0101 04050030 6E310B30
```

```
  09060355 04061302 55533110 300E0603 55040813 0747656F 72676961 31183016
  06035504 0A130F54 72616E73 4E657875 732C204C 4C433114 30120603 55040B13
  0B446576 656C6F70 6D656E74 311D301B 06035504 03131454 52414E53 4E455855
  53204245 54412043 41203130 1E170D39 39303332 32313334 3630395A 170D3030
  30333231 31333436 30395A30 6E310B30 09060355 04061302 55533110 300E0603
  55040813 0747656F 72676961 31183016 06035504 0A130F54 72616E73 4E657875
  732C204C 4C433114 30120603 55040B13 0B446576 656C6F70 6D656E74 311D301B
  06035504 03131454 52414E53 4E455855 53204245 54412043 41203130 819F300D
  06092A86 4886F70D 01010105 0003818D 00308189 02818100 B1B8ACFC D78F0C95
  0258D164 5B6BD8A4 6F5668BD 50E7524B 2339B670 DC306537 3E1E9381 DE2619B4
  4698CD82 739CB251 91AF90A5 52736137 658DF200 FAFEFE6B 7FC7161D 89617E5E
  4584D67F F018EDAB 2858DDF9 5272F108 AB791A70 580F994B 4CA54F08 38C32DF5
  B44077E8 79830F95 96F1DA69 4CAE16F2 2879E07B 164F5F6D 02030100 01300D06
  092A8648 86F70D01 01040500 03818100 2FDCB580 C29E557C 52201151 A8DB5F47
  C06962D5 8FDA524E A69DE3EE C3FE166A D05C8B93 2844CD66 824A8859 974F22E0
  46F69F7E 8027064F C19D28BC CA750E4E FF2DD68E 1AA9CA41 8BB89C68 7A61E9BF
  49CBE41E E3A42B16 AAEDAEC7 D3B4F676 4F1A817B A5B89ED8 F03A15B0 39A6EBB9
  0AFA6968 17A9D381 FD62BBB7 A7D379E5
  quit
 certificate 8697B659C0E190E1A8D48961EBED0DB1
  30820247 308201B0 A0030201 02021100 8697B659 C0E190E1 A8D48961 EBED0DB1
  300D0609 2A864886 F70D0101 04050030 6E310B30 09060355 04061302 55533110
  300E0603 55040813 0747656F 72676961 31183016 06035504 0A130F54 72616E73
  4E657875 732C204C 4C433114 30120603 55040B13 0B446576 656C6F70 6D656E74
  311D301B 06035504 03131454 52414E53 4E455855 53204245 54412043 41203130
  1E170D39 39303430 36313833 3430315A 170D3030 30343036 31383334 30315A30
  81873181 84300F06 03550405 13083131 38313833 37393018 06092A86 4886F70D
  01090813 0B312E31 342E3131 352E3835 302A0609 2A864886 F70D0109 02161D61
  73353330 302D3035 2E666965 6C646C61 62732E63 6973636F 2E636F6D 302B0603
  55040314 245B7472 616E736E 65787573 2E636F6D 20475749 443D3230 30302043
  5349443D 31303030 5D305C30 0D06092A 864886F7 0D010101 0500034B 00304802
  4100AF40 5CC8E37D 7211E3C4 2D036E52 70B5DA88 96600C12 8654B85E 7CEFE204
  27A9B9DD B0F6B85C 1EB561BB 0F3481A2 D4661087 2B0B403A 5A65B7E0 ED9A0165
  EBC10203 010001A3 0F300D30 0B060355 1D0F0404 030205A0 300D0609 2A864886
  F70D0101 04050003 8181005C 1E379447 C0FCBC3F 0ABC75FA ADF79A26 770419A4
  02BEC849 ECB7BDB1 58EA815B 48844DB3 4E8934E8 397F4762 F04EB716 8413C418
  4289AA64 6E2EAFE1 9C9F1F31 3A5BE996 AF749623 18FBFD36 569732BF 8335C522
  4ACA0BCA CFCC27C6 294AD416 15472F07 C1609E93 E1FEDA66 B69DA603 1A99699E
  86937EC5 609A3D52 72A45B
  quit
!
xgcp snmp sgcp
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 clock source line secondary 1
!
controller T1 2
!
controller T1 3
!
voice-port 0:D
!
dial-peer voice 1 pots
 application session
 destination-pattern 5710877
 port 0:D
!
dial-peer voice 5 voip
```

```
 application session
 incoming called-number +1404.......
 session target settlement:0
!
dial-peer voice 2 pots
 destination-pattern +255....
 port 0:D
 prefix 255
!
! Enable roaming for this dialpeer
!
dial-peer voice 6 voip
 roaming
 destination-pattern 1512.......
 session target settlement
!
dial-peer voice 7 pots
 destination-pattern +1650.......
 port 0:D
 prefix 1650
!
dial-peer voice 8 voip
 application session
 incoming called-number +1650.......
 session target settlement:0
!
dial-peer voice 3 voip
 application session
 incoming called-number +1408.......
 session target settlement:0
!
dial-peer voice 12 pots
 destination-pattern 1404.......
 port 0:D
 prefix 1404
!
dial-peer voice 13 pots
 destination-pattern 1512.......
 port 0:D
 prefix 1512
!
!User with account number matching 875.... is a roaming caller
!
settlement roam-pattern 875.... roam
!
!Enable roaming for this settlement provider using the "roaming" attribute
!
settlement 0
 type osp
 url https://1.14.115.100:8443/
 device-id 2000
 customer-id 1000
 roaming
 no shutdown
!
!
interface Ethernet0
 ip address 1.14.115.85 255.255.0.0
 no ip directed-broadcast
 no ip mroute-cache
 no cdp enable
!
interface Serial0:23
 no ip address
```

```
 no ip directed-broadcast
 dialer-group 1
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 no ip address
 no ip directed-broadcast
 no ip mroute-cache
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
router igrp 200
 network 1.0.0.0
!
ip default-gateway 1.14.0.1
ip classless
ip route 172.16.0.0 255.255.0.0 1.14.115.65
no ip http server
!
no cdp run
!
line con 0
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 password lab
 login
!
scheduler interval 1000
end
```

Settlement with PKI Multiple Roots Example

The following example shows configuration of settlement with PKI Multiple Roots on the settlement server. As shown in the example, the router has been enrolled under VeriSign TestDerive CA. It has confided Netscape CMS as a trusted root. The Netscape CMS is installed on the server Cisco ca-ultra.

```
version 12.2
service timestamps debug datetime
service timestamps log datetime
no service password-encryption
service internal
!
hostname as5300-04
!
enable secret 5 $1$Ld7z$CapnZCfz2kMSh8sMHh2hy0
enable password lab1
!
resource-pool disable
!
ip subnet-zero
ip domain-name fieldlabs.cisco.com
ip name-server 172.16.2.132
!
```

```
isdn switch-type primary-5ess
isdn voice-call-failure 0
cns event-service server
mta receive maximum-recipients 1024
!
crypto cisco algorithm des
crypto cisco algorithm des cfb-8
crypto cisco algorithm 40-bit-des
!
!Configure the second root to be downloaded from tftp server
!
crypto ca trusted-root transnexus2
 root tftp 1.14.115.100 onsite_ca.der
!
crypto ca identity transnexus
 enrollment retry count 100
 enrollment retry period 2
 enrollment url http://hostname
crypto ca certificate chain transnexus
 certificate ca 0171
  3082024C 308201B5 02020171 300D0609 2A864886 F70D0101 04050030 6E310B30
  09060355 04061302 55533110 300E0603 55040813 0747656F 72676961 31183016
  06035504 0A130F54 72616E73 4E657875 732C204C 4C433114 30120603 55040B13
  0B446576 656C6F70 6D656E74 311D301B 06035504 03131454 52414E53 4E455855
  53204245 54412043 41203130 1E170D39 39303332 32313334 3630395A 170D3030
  30333231 31333436 30395A30 6E310B30 09060355 04061302 55533110 300E0603
  55040813 0747656F 72676961 31183016 06035504 0A130F54 72616E73 4E657875
  732C204C 4C433114 30120603 55040B13 0B446576 656C6F70 6D656E74 311D301B
  06035504 03131454 52414E53 4E455855 53204245 54412043 41203130 819F300D
  06092A86 4886F70D 01010105 0003818D 00308189 02818100 B1B8ACFC D78F0C95
  0258D164 5B6BD8A4 6F5668BD 50E7524B 2339B670 DC306537 3E1E9381 DE2619B4
  4698CD82 739CB251 91AF90A5 52736137 658DF200 FAFEFE6B 7FC7161D 89617E5E
  4584D67F F018EDAB 2858DDF9 5272F108 AB791A70 580F994B 4CA54F08 38C32DF5
  B44077E8 79830F95 96F1DA69 4CAE16F2 2879E07B 164F5F6D 02030100 01300D06
  092A8648 86F70D01 01040500 03818100 2FDCB580 C29E557C 52201151 A8DB5F47
  C06962D5 8FDA524E A69DE3EE C3FE166A D05C8B93 2844CD66 824A8859 974F22E0
  46F69F7E 8027064F C19D28BC CA750E4E FF2DD68E 1AA9CA41 8BB89C68 7A61E9BF
  49CBE41E E3A42B16 AAEDAEC7 D3B4F676 4F1A817B A5B89ED8 F03A15B0 39A6EBB9
  0AFA6968 17A9D381 FD62BBB7 A7D379E5
  quit
 certificate B7DD210B9BFE007E41EEB177AF39F78C
  30820247 308201B0 A0030201 02021100 B7DD210B 9BFE007E 41EEB177 AF39F78C
  300D0609 2A864886 F70D0101 04050030 6E310B30 09060355 04061302 55533110
  300E0603 55040813 0747656F 72676961 31183016 06035504 0A130F54 72616E73
  4E657875 732C204C 4C433114 30120603 55040B13 0B446576 656C6F70 6D656E74
  311D301B 06035504 03131454 52414E53 4E455855 53204245 54412043 41203130
  1E170D39 39303430 36313833 3635325A 170D3030 30343036 31383336 35325A30
  81873181 84300F06 03550405 13083131 37363837 37353018 06092A86 4886F70D
  01090813 0B312E31 342E3131 352E3834 302A0609 2A864886 F70D0109 02161D61
  73353330 302D3034 2E666965 6C646C61 62732E63 6973636F 2E636F6D 302B0603
  55040314 245B7472 616E736E 65787573 2E636F6D 20475749 443D3130 30302043
  5349443D 31303030 5D305C30 0D06092A 864886F7 0D010101 0500034B 00304802
  4100C82B 8E4CBD44 06C763FB 1DC1A78F 8D71F1DA 110EDAC3 C9AA6256 6E1BF15B
  79E48BEF 741D26CF DEBEACCC FA09D420 F54B76A1 F6CDCE33 02C8D9F7 5873E012
  AFC90203 010001A3 0F300D30 0B060355 1D0F0404 030205A0 300D0609 2A864886
  F70D0101 04050003 81810056 C05E1151 BE2D5515 624010AE 22F03D58 8BD9F2D3
  E037EBC8 376E321A 5C53D4C6 770CE32F CF1CB0F4 2FD44C0D CA8EE22C 2372EE64
  349FF062 137A6780 DC554F6A 3BA9F17C 85A7F390 D5B99E35 D7FBF927 75910E9E
  992C7052 54AE0887 ED1DEEA0 C6BCA9C4 49F3D98E 4835A5E2 0FD470B6 F6D727A8
  8AA0F923 5D60985B F8DD19
  quit
crypto ca certificate root transnexus2 DB3882D37891B597970BF0F18B008F13
 308201F4 3082015D A0030201 02021100 DB3882D3 7891B597 970BF0F1 8B008F13
 300D0609 2A864886 F70D0101 04050030 15311330 11060355 040A130A 5472616E
```

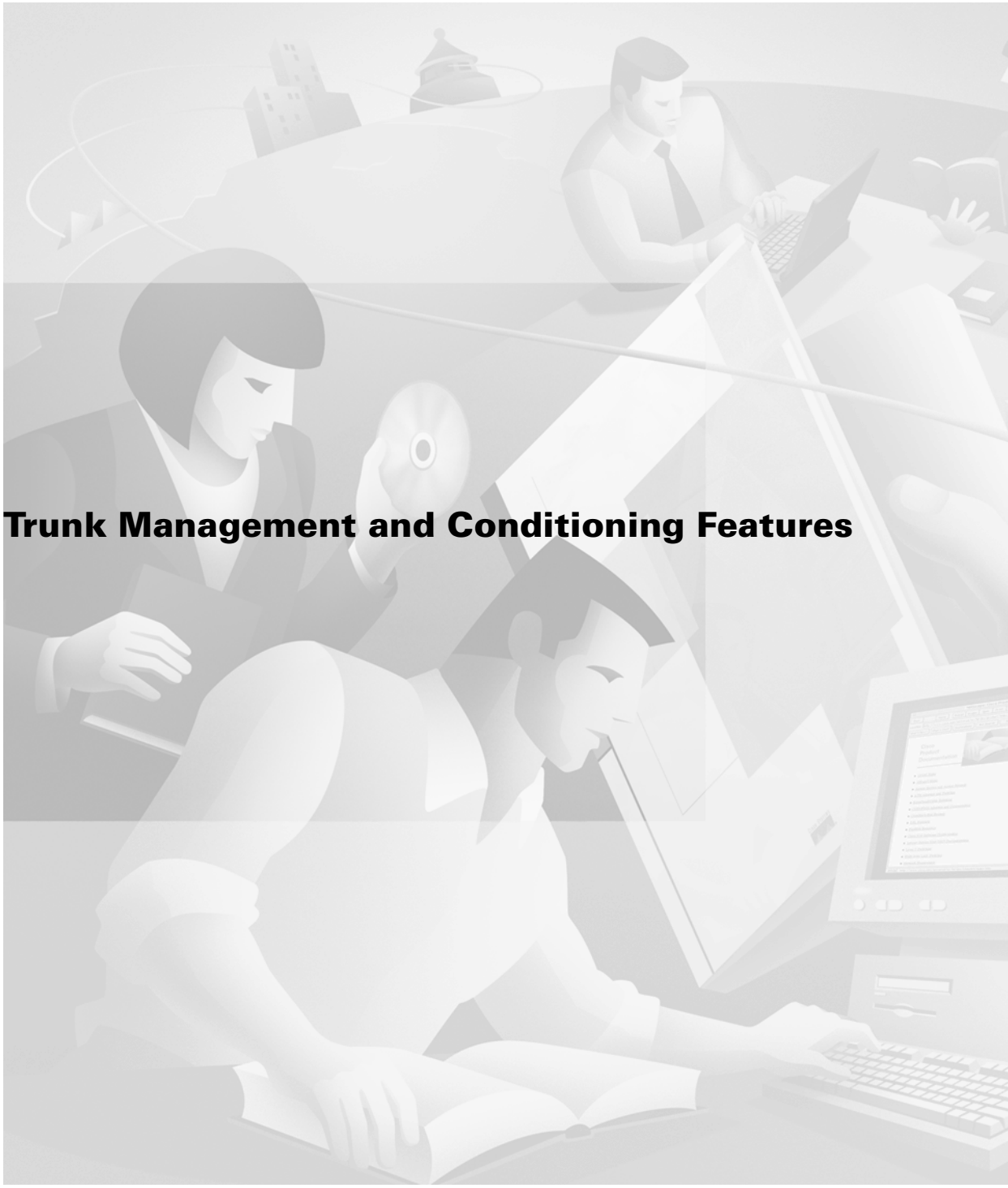
```
 734E6578 7573301E 170D3939 30333138 30303030 30305A17 0D303930 33313832
 33353935 395A3015 31133011 06035504 0A130A54 72616E73 4E657875 7330819F
 300D0609 2A864886 F70D0101 01050003 818D0030 81890281 8100AB91 E2123C3F
 E83DE86A 3B8A18DF 750FB756 3034D692 2A363692 721F9E59 6CDB046F AAF9A212
 6B4B1033 9DDE94DB B132E768 085376EC 9EC7E2FD 0BB92B43 8FEC1243 35A33F89
 41390517 AF2D6D46 2FAAC116 8AE55865 C326C77A 3381C944 5BE107B1 E66CA111
 B3560313 A29A0081 201D84C5 FE24E452 6338C52C EFDE6B95 4A570203 010001A3
 44304230 22060355 1D11041B 3019A417 30153113 30110603 55040313 0A4F6E73
 69746532 2D363230 0F060355 1D130408 30060101 FF020100 300B0603 551D0F04
 04030201 06300D06 092A8648 86F70D01 01040500 03818100 481E4F13 79EB3B5F
 D9BCEED9 9C756BF7 B42167B1 4DE11B8C 240D3446 5A14E2E1 A79D2454 1EA84109
 17EF6E8E 8AFD06C7 8209753B F760761C EC13A2D6 95348D69 4F73F0D5 9211DD95
 0FE00D23 4583002A 242C769E 695FAFD4 EE12D014 580C5DFC F377F3FF F20F25D6
 831E4F2B 253DFA9C 8B3E00A8 002F03D7 BC0C19D8 7EA134A6
 quit
!
xgcp snmp sgcp
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 clock source line secondary 1
!
controller T1 2
!
controller T1 3
!
voice-port 0:D
!
dial-peer voice 1 pots
 application session
 destination-pattern 5710876
 port 0:D
!
dial-peer voice 7 voip
 destination-pattern +255....
 session target settlement:0
!
dial-peer voice 13 pots
 destination-pattern 1770.......
 port 0:D
 prefix 1770
!
dial-peer voice 1770 voip
 incoming called-number 1770.......
 ip precedence 7
 session target settlement:0
!
dial-peer voice 1650 voip
 destination-pattern +1650.......
 session target settlement:0
!
dial-peer voice 10 voip
 destination-pattern 1408.......
 session target settlement
!
dial-peer voice 1404 voip
 destination-pattern 1404.......
 session target settlement
!
```

```
dial-peer voice 1512 voip
 destination-pattern 1512.......
 session target settlement
!
!Specify which root to use to validate the settlement token
!via token-root-name attribute
!
settlement 0
 type osp
 url https://1.14.115.100:8443/
 retry-delay 2
 device-id 1000
 customer-id 1000
 token-root-ca transnexus2
 no shutdown
!
interface Ethernet0
 ip address 1.14.115.84 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no cdp enable
!
interface Serial0:23
 no ip address
 no ip directed-broadcast
 dialer-group 1
 isdn switch-type primary-5ess
 isdn protocol-emulate user
 isdn incoming-voice modem
 fair-queue 64 256 0
 no cdp enable
!
interface FastEthernet0
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
 no cdp enable
!
router igrp 200
 network 1.0.0.0
!
ip default-gateway 1.14.0.1
ip classless
no ip http server
!
no cdp run
!
line con 0
 logging synchronous
 transport input none
line aux 0
line vty 0 4
 password lab
 login
!
ntp clock-period 17180879
ntp update-calendar
ntp server 1.14.42.23
scheduler interval 1000
end
```

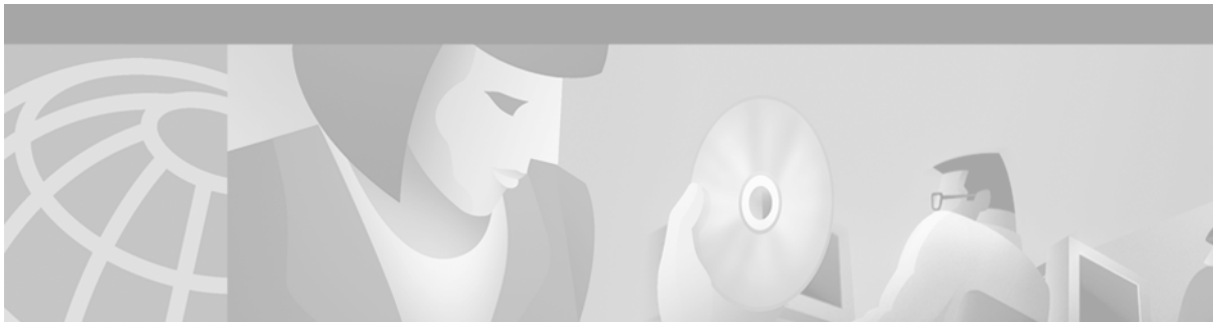
Settlement with UNI-OSP Example

The following configuration example shows UNI-OSP settlement:

```
Router# settlement 0
Router# type uni-osp
Router# url 172.16.100.1
```



Trunk Management and Conditioning Features



Configuring Trunk Connections and Conditioning Features

This chapter describes trunk connections and conditioning features for the Cisco 2600 and 3600 series routers and MC3810 multiservice concentrators. The features include trunk conditioning, tie-line simulation, T1/E1 alarms, Public Switched Telephone Network (PSTN) fallback, and busyout. This chapter contains:

For a complete description of the commands in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

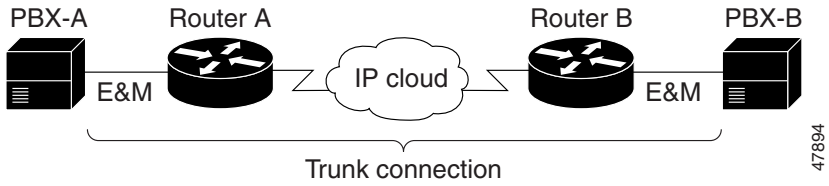
Trunking Overview

A trunk is a communication line between two switching systems—the switching equipment in a central office (CO) and PBX. It is a physical and logical point-to-point connection with a permanent wire over which network traffic travels. A backbone is composed of a number of trunks.

Voice over IP (VoIP) simulates trunk connections. The simulated connections occur between PBXs that are connected to Cisco routers or access servers on each side of the network

In Figure 120, two PBXs are connected to a router using a simulated trunk and a recEive and transMit (E&M) voice port. In this case, a permanent, non-switched connection transparently connects the two PBXs.

Figure 120 Simulated Trunk Connection



Simulated Lines and Trunks

Simulated lines and trunks enable a telephone user at one location to dial an access code to access a PBX at another location. A second dial tone can be heard coming from the remote PBX. There are two types of simulated connections—*switched* and *permanent*—that can be configured for both analog and digital systems. The connections are created with the Cisco **connection** command.

The connection trunk command creates a permanent call that is connected as soon as the routers on each end are booted (see Figure 121). Permanent calls pass limited telephony signaling and operate without collecting digits or requiring changes to the overall dial plan.

Figure 121 Connection Trunk Configuration

The calls simulate a permanent tie-line between two PBXs. Both ends must be configured and have compatible voice-port signaling that is:

- E&M to E&M
- Foreign Exchange Office (FXO) to Foreign Exchange Station (FXS)

The signaling cannot be FXO to ground start.

When a switched call is configured (see Figure 122), the user can make a call without dialing any digits. The telephony signaling, such as hookflash, is not passed. The call will not roll over to voice mail if the remote telephone does not answer and digits from an attached telephony device are not collected.

Figure 122 Connection Private-Line Auto Ringback (PLAR) Configuration

The switched call configuration works with any type of voice port (E&M, FXO, or FXS) and can be used without any effect on an existing dial plan. It is commonly used to connect PBXs in which the remote devices appear to be physical extensions. The PBX provides dial tone to the extensions, not the router.

The **connection tie-line** command creates a switched call between two stations or PBXs that bypasses the switch. The **connection plar-opx** command creates a call that is similar to a switched call. The connection does not take place between the PBX and the local router until the far-end FXS device answers. This enables the PBX to provide centralized voice mail or attendant services when the remote device does not answer.

Trunk Conditioning Signaling Attributes

Trunk conditioning signaling attributes apply to permanent point-to-point voice connections (private lines and tie-lines) created using the **connection trunk** command. This feature provides the following capabilities:

- Creation of voice classes.
- Specific signaling attributes in each voice class.
- Signaling attributes in the voice class for Voice over Frame Relay (VoFR) and Voice over Asynchronous Transfer Mode (VoATM) dial peers.

Trunk conditioning enables greater control over Cisco private-line calls that are sent over Frame Relay or ATM networks. When private-line or tie-line calls are sent between two PBXs, fault indications are sent to the sending PBX. If the call fails, the PBX is able to select an alternate path to route the calls. Selecting an alternate path applies to analog connections or digital T1/E1 using channel-associated signaling (CAS)/robbed-bit ABCD signaling. It does not cover common channel signaling (CCS).

When T1/E1 CAS is carried in transparent pass-through mode for arbitrary, unknown, or unsupported CAS protocols, it is necessary to define on-hook/idle patterns so that the domain specific part (DSP)/signaling code can sense the idle call state and shut off the flow of voice packets when no active call is in progress. This mode provides an additional idle bandwidth-saving mechanism for those cases when Voice Activity Detection (VAD) is not desired.

Note Cisco MC3810 series concentrators support additional trunk-conditioning features that specify timing, signaling, and transmission options. The features provide enhanced control over call rerouting in cases of trunk failure and increased bandwidth availability due to suppression of voice packets on Out-of-Service (OOS) trunks.

Congestion Monitoring and Management Features

Congestion monitoring of permanent and switched calls is performed with these features: T1/E1 alarm conditioning, PSTN fallback, and busyout functionality including busyout monitoring. These features provides the following capabilities:

- Signaling and suppression of voice traffic for idle or OOS network trunks.
- Busyout of the ports interfacing with a local PBX.

An OOS condition can be signalled using an ABCD bit pattern that is different from the busy or seized state. The difference enables the PBX to differentiate between OOS and congestion.

T1/E1 Alarm Conditioning

Alarm conditioning provides status monitoring on T1/E1 PBX voice interfaces for simulated lines and trunks created using the **connection** command. It supports operation with CAS, but does not support CCS.

A T1/E1 alarm can be triggered by events detected through the monitoring of a specified set of voice ports within a T1/E1 trunk. A monitored set includes a defined voice port that has a specified DS0 group or groups and configured for one of the following:

- End-to-end connection of permanent virtual circuits (PVCs)
- Busyout of switched virtual circuits (SVCs), where the busyout state is initiated using the **busyout monitor** command.

When all the monitored voice ports on a T1/E1 trunk are OOS (PVCs are OOS and SVCs are busied out), a T1/E1 Alarm Indication Signal (AIS) is generated on the T1/E1 trunk connected to the PBX or PSTN.

Note Voice ports busied out by the **busyout forced** command do not trigger a T1/E1 alarm.

PSTN Fallback

PSTN fallback monitors congestion in the IP network and redirects calls to the PSTN or reject calls based on the network congestion. PSTN fallback is supported on Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrators. For information concerning Voice over IP (VoIP), Voice over ATM (ATM), Calculated Impairment Planning Factor (ICPIF), and Service Assurance Agent (SAA), see the following:

- *Cisco IOS Multiservice Applications Configuration Guide*
- *Cisco IOS Multiservice Applications Command Reference*
- Configuring Voice over ATM for the Cisco MC3810
- Voice over ATM on Cisco 3600 Series Routers
- *Managing Voice Quality with Cisco Voice Manager (CVM) and Telemate*
- *Monitoring the Router and Network*

PSTN fallback can re-routed calls to an alternate IP destination or to the PSTN if the IP network is found unsuitable for voice traffic at that time. The user defines the congestion thresholds based on the configured network. This functionality enables the service provider to give a reasonable guarantee about the quality of the conversation to their VoIP users at the time of call admission.

Note PSTN fallback does not ensure that a VoIP call is protected from the effects of congestion. This is the function of the other Quality of Service (QoS) mechanisms such as IP Real-Time Transport Protocol (RTP) priority or low latency queueing (LLQ).

PSTN fallback includes the following features:

- Offers flexibility to define the congestion thresholds based on the network by:
  - Defining a threshold based on ICPIF, which is derived as part of International Telecommunication Union (ITU) G.113.
  - Defining a threshold based solely on packet delay and loss measurements.

- Uses SAA probes to provide packet delay, jitter, and loss information for the relevant IP addresses. Based on the packet loss, delay, and jitter encountered by these probes, an ICPIF or delay/loss value is calculated. See "Service Assurance Agent" section on page 576.
- Supports calls of any codec. Only G.729 and G.711 have accurately simulated probes. Calls of all other codecs are emulated by a G.711 probe.

The fallback subsystem has a network traffic cache that maintains the ICPIF or delay/loss values for various destinations. The subsystem helps performance, because new calls to a well-known destination do not have to wait on a probe. The value is usually cached from a previous call.

Once the ICPIF or delay/loss values are calculated and stored, the values remain until the cache ages out or overflows. Until an entry ages out, probes are sent periodically for that destination. The time interval is user configurable. In the following example, it is assumed that call fallback active is enabled and an ICPIF threshold is defined. The call control would be similar if loss and delay thresholds were defined.

Step 1 A call comes into the router. The IP address of the destination is checked against the configured maps to see if it should be sent to another router, such as a backhaul router, or to an alternate dial peer. If it should be sent to another router, the IP address for the fallback subsystem is replaced with the target router. If it should be sent to an alternate dial peer, the router matches that dial peer and obtains the destination information (codec, IP address, and so on).

Note The change is made in the destination address of the probing address. The destination for the actual call is not changed.

Step 2 The router calls the fallback subsystem to look up the specified destination in its network traffic cache. If the ICPIF value exists and is current, then the router uses that value to decide whether to permit the call into the VoIP network. If the router determines that the network congestion is below the configured threshold (by looking at the value from the probe or a cached value), then the call is connected. Otherwise, the router checks the next dial-peer match again in the same way. Eventually, if all the VoIP dial peers are deemed unsuitable, then the call is hairpinned to the PSTN by virtue of a configured POTS dial peer (for analog or digital interfaces). If no PSTN dial peer is present, a fast-busy is sent to the PBX (in case of digital interfaces).

Note It is not possible to signal a fast-busy to some interfaces.

Step 3 The fallback subsystem continues probing in the background periodically (period time is configured by the **call fallback probe-timeout** command), so that the network congestion information is available when there is a call request. The first call for a particular dial peer may be delayed while the router calculates the congestion information for that destination.

If the timeout threshold is set and the router has not received calls for a particular destination after the threshold expires, then the router removes that destination's traffic information from the cache.

Calculated Impairment Planning Factor

ICPIF calculates an impairment factor for every piece of equipment along the voice path and adds the values to get the total impairment. The ITU assigns the different types of impairments, such as noise, delay, and echo.

The ICPIF handling has been introduced for compatibility with Cisco H.323. Part of ICPIF includes a concept of Total Impairment Value that is a function of loss of packets, delay of packets, and codecs used based on the round-trip reports from SAA. For this feature, all codecs are classified as 729 class codecs or 711 class codecs.

Service Assurance Agent

SAA is a network congestion analysis mechanism. SAA provides delay, jitter, and packet loss information for the configured IP addresses. SAA is based on a client-server protocol defined on UDP. It has an Message Digest 5 (MD5), which is a message authentication algorithm in SNMP v.2. MD5 verifies the integrity of the communication, authenticates the origin, and checks for timeliness.

SAA uses UDP port (port 1976) for sending the SAA control message to the terminating gateway. The SAA probe packets go out on randomly selected ports from the top end of the audio UDP port range (16384 - 32767).

The port pair (RTP & Real-Time Transport Control Protocol [RTCP] port) is selected, and by default SAA for call fallback uses the RTCP port (odd number) to avoid going into the priority queue, if enabled. If fallback is configured to use the priority queue, the RTP port (even number) is selected. The audio UDP port range must be included in the priority queue for fallback priority queueing to work.

Busyout

Three busyout conditions are discussed in the following sections:

- Local Voice Busyout, page 576
- Advanced Voice Busyout, page 577
- Busyout Monitor, page 577

Local Voice Busyout

Local voice busyout is designed to busy out trunks assigned to PVCs so that the PBX does not seize the circuit. Local voice busyout enables the PBX to route a call based on the actual availability of trunks. Local voice busyout enables the following:

- A group of voice ports to be marked busy if a link is broken.
- Specific voice ports in a PVC application to be marked busy under specified conditions.

When ports are marked busy, a call is forced back to the originating equipment (typically a PBX) that reroutes the call over an alternate path. This action ensures that a caller does not experience "dead air" resulting from a connection that never terminates.

The local voice busyout feature provides a way to busy out a voice port if a monitored network interface changes state. When a monitored interface changes to a specified state—to OOS or in-service—the voice port presents a seized/busyout condition to the attached PBX or other customer premises equipment (CPE). The PBX or other CPE can then attempt to select an alternate route.

Local voice busyout is different from busy-back. *Busy-back* refers to the signal sent from within the network to the calling party that indicates a busy (or congested) state anywhere along the route, up to and including the condition of the called party.

Note Local voice busyout is supported on analog and digital voice ports using CAS, but not on BRI Voice Modules (BVMs).

Advanced Voice Busyout

Advanced voice busyout monitors links to remote and IP-addressable interfaces and uses an SAA probe signal for VoIP. Voice classes are configured to simplify and speed up the configuration of voice busyout on multiple voice ports. SAA probe monitoring of remote interfaces is intended for use with VoIP, VoFR, and VoATM networks.

Busyout Monitor

Busyout monitor is one aspect of Call Admission Control (CAC) that uses a data network and the PSTN to provide the best possible quality and cost savings for VoIP calls. Busyout monitor CAC functionality also provides the following:

- Logical connections between LAN/WAN interfaces of routers in a VoIP gateway with directly connected voice ports.
- Port-by-port definition.
- Tracking of any directly connected main interface, subinterface, or virtual interface without monitoring the status of remote devices.

Trunk Management Prerequisite Tasks

Before configuring the trunk connections and conditioning features, the one of the following must be configured:

- VoFR using FRF.11
- VoATM
- VoIP
- Voice ports

Before configuring the congestion-monitoring features, the following requirements must be met:

- Alarm conditioning requires Cisco IOS Release 12.1(3)T or later. The following must also be configured:
  - VoFR or VoATM, including plain old telephone service (POTS) and network dial peers
  - Voice ports, including busyout and trunk conditioning
  - DS0 groups
- PSTN fallback requires that VoIP be configured.
- Voice busyout and SAA probe enhancements required that the following configuration tasks be completed:
  - VoFR or VoATM, including POTS and network dial peers
  - Voice ports
  - VoIP network
  - Call fallback on the local router
  - SAA responder on the target (far-end) router

Note Trunk Management Configuration Tasks List

This section includes procedures for configuring the following trunk management features:

- Configuring Trunk-Conditioning Signaling Attributes, page 578
- Assigning Trunk-Conditioning Attributes to Network Dial Peers, page 581
- Assigning Voice Classes to Voice Ports, page 582
- Configuring Trunk Connections, page 584
  - Configuring PLAR (Switched) Connections, page 584
  - Configuring Trunk/Tie-Line Connections, page 585
  - Configuring PLAR-OPX Connections, page 589
- Configuring T1/E1 Alarm Generation Parameters, page 589
- Configuring PSTN Fallback, page 592
- Configuring Local Voice Busyout, page 595

Configuring Trunk-Conditioning Signaling Attributes

Different trunk-conditioning signaling attributes may be required to match the characteristics of the different PBXs to which the router connects. For this reason, trunk-conditioning attributes are configured by creating a voice class for each set of attributes required. The trunk-conditioning attributes are configured for the voice class and the voice class is assigned to one or more dial peers.

A voice class must be configured and assigned to at least one dial peer before the trunk conditioning signaling attributes take effect.

Note This configuration supports the North America CAS Protocol and applies only to Cisco private-line or FRF.11 trunk calls. It does not apply to digital T1/E1 trunks using CCS.

To create a voice class and define the trunk-conditioning attributes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class permanent` *tag*	Creates a voice class. The *tag* number range is from 1 to 10000, and it must be unique on the router. **Note** The **voice-class** command in dial-peer configuration mode is entered with a hyphen. The **voice class** command in global configuration mode is entered without the hyphen.
Step 2	`Router(config-voice-class)# signal keepalive` *seconds*	(Optional) Defines the keepalive signaling packet interval. The *seconds* range is from 1 to 65535; the default is 5.
Step 3	`Router(config-voice-class)#` {**no-action** | **idle-only** | **oos-only** | **both**}	(Optional) Sets the signaling pattern (when the far-end keepalive message is lost or when AIS is received from the far end). The keywords are as follows: • **no-action**—Sends no signaling pattern. • **idle-only** or **oos-only**—Sends only one signaling pattern. • **both**—Restores the default (both signaling patterns are sent). **Note** The **no** form of the command restores the default also.
Step 4	`Router(config-voice-class)# signal pattern` {**idle receive** | **idle transmit** | **oos receive** | **oos transmit**} *bit-pattern*	(Optional) Overrides the default values for the idle and receive OOS patterns or configures OOS transmit signaling patterns. The keywords and argument are as follows: • **idle receive**—Defines the signaling pattern for an idle message from the network and the signaling pattern to be sent to the PBX if the network trunk is OOS and **signal sequence oos idle-only** or **signal sequence oos** are configured. The defaults are: – For near-end E&M—0000 (for T1) or 0001 (for E1) – For near-end FXO loop start—0101 – For near-end FXO ground start—1111 – For near-end FXS—0101 – For near-end MELCAS—1101

Command	Purpose
	• **idle transmit**—Defines the signaling pattern for an idle message from the PBX. The defaults are: – For near-end E&M—0000 – For near-end FXO—0101 – For near-end FXS loop start—0101 – For near-end FXS ground start—1111 – For near-end MELCAS—1101 • **oos receive**—Defines the OOS signaling pattern to be sent to the PBX if the network trunk is OOS and **signal sequence oos oos-only** or **signal sequence oos** are configured. The defaults are: – For near-end E&M—1111 – For near-end FXO loop start—1111 – For near-end FXO ground start—0000 – For near-end FXS loop start—1111 – For near-end FXS ground start—0101 – For near-end MELCAS—1111 • **oos transmit**—Defines the signaling pattern for an OOS message from the PBX. There are no default signaling patterns defined. • *bit-pattern*—Defines the ABCD bit pattern. Valid values are from 0000 to 1111. The receive signal pattern comes from the data network side to the PBX. The transmit signal pattern comes from the PBX to the data network side. The range for all options is from 0000 to 1111. Repeat the command entry for each signal pattern required.
Step 5 `Router(config-voice-class)# signal timing oos timeout {`*seconds* `| disabled}`	(Optional) Changes the timeout period for asserting a receive OOS pattern to the PBX when signaling packets are lost. This action changes the delay time before a busyout is sent to the PBX. The keyword and argument are as follows: • *seconds*—Defines the delay duration between the loss of signaling packets and the beginning of the OOS state. The range is from 1 to 65535. The default is 30. • **disabled**—Deactivates the detection of packet loss. If no signaling packets are received from the network, the router does not send an OOS pattern to the PBX and it continues sending voice packets. Use this option to disable busyout to the PBX.

	Command	Purpose
Step 6	`Router(config-voice-class)# signal timing oos restart` *seconds*	(Optional) Configures permanent voice connections to be restarted after the trunk has been OOS for a specified time. The default is no signal timing OOS pattern parameters are configured. **Note** This command has no effect if **signal timing oos timeout** is set to **disabled**.
Step 7	`Router(config-voice-class)# signal timing oos slave-standby` *seconds*	(Optional) Configures a slave port to return to its initial standby state after the trunk has been OOS for a specified time. The default is no signal timing OOS pattern parameters are configured. **Note** This command has no effect if **signal timing oos timeout** is set to **disabled**.
Step 8	`Router(config-voice-class)# signal timing oos {suppress-all | suppress-voice}` *seconds*	(Optional) Configures the router or concentrator to stop sending voice packets or voice and signaling packets to the network if it detects a transmit OOS signaling pattern from the PBX for a specified time. The default is no signal timing OOS pattern parameters are configured. **Note** An OOS transmit signaling pattern must be configured with the **signal pattern oos transmit** command (see Step 4).
Step 9	`Router(config-voice-class)# signal timing idle suppress-voice` *seconds*	(Optional) Configures the router or concentrator to stop sending voice packets after the trunk has been idle for a specified time. The default is no signal timing OOS pattern parameters are configured.

Assigning Trunk-Conditioning Attributes to Network Dial Peers

After the voice class has been created, it must be applied to the dial-peer configuration. The trunk-conditioning attributes can be assigned to VoIP, VoFR, or VoATM dial peers, but not to POTS dial peers.

Note This feature applies only to Cisco trunk (private-line) or FRF.11 trunk calls and does not apply to digital T1/E1 trunks using CCS.

To apply trunk-conditioning signaling attributes to a network dial peer, specify the dial peer type and then use the following command in dial-peer voice configuration mode:

Command	Purpose
`Router(config-dial-peer)# voice-class permanent` *tag*	Assigns the voice class to the dial peer. The *tag* argument specifies the unique number. The valid range is from 1 to 10000. **Note** The **voice-class** command in dial-peer configuration mode is entered with a hyphen. The **voice class** command in global configuration mode is entered without the hyphen.

Assigning Voice Classes to Voice Ports

To assign a voice class to a voice port, specify the voice port, and then use the following command in voice-port configuration mode:

Command	Purpose
`Router(config-voice-port)# voice-class permanent` *tag*	Assigns the voice class to a voice port. The *tag* argument is a unique number assigned to the voice class. Valid range is from 1 to 10000. **Note** The **voice-class** command for assigning a voice class to a voice port has a hyphen. The **voice class** command in global configuration mode is entered without the hyphen.

Verifying the Signaling Attributes and Trunk Conditioning

To verify the signaling attributes (timing parameters) using voice-port 1/5 on a Cisco MC3810 multiservice concentrator, enter the **show voice trunk-conditioning signaling** command. The following is a sample output from this command:

```
Router# show voice trunk-conditioning signaling 1/5

1/5 :
TX INFO :slow-mode seq#= 25, sig pkt cnt= 42, last-ABCD=0000
hardware-state ACTIVE signal type is NorthamericanCAS
signal path is OPEN
 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
RX INFO :slow-mode, sig pkt cnt= 37
missing = 0, out of seq = 0, very late = 0
playout depth = 0 (ms), refill count = 1
prev-seq#= 25, last-ABCD=0000
trunk_down_timer = 4212 (ms), idle timer = 0 (sec),
tx_oos_timer = 0 (sec), rx_ais_duration = 0 (ms)
forced playout signal pattern = NONE
signaling playout history
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

To verify the status of trunk supervision and configuration parameters on a Cisco MC3810 multiservice concentrator, enter the **show voice trunk-conditioning supervisory** command. The following is a sample output from this command.

```
Router# show voice trunk-conditioning supervisory 1/5

1/5 : state : TRUNK_SC_CONNECT, voice : on, signal : on, slave
status: trunk connected
sequence oos : idle and oos
pattern :rx_idle = 0x0 rx_oos = 0xF tx_oos =  0xF
timing : idle = 0, restart = 0, standby = 0, timeout = 40
supp_all = 50, supp_voice = 0, keep_alive = 5
timer: oos_ais_timer = 0, timer = 0
```

To verify signaling and timing parameters for the configuration for voice-ports 0:0, 0:1, and 0:2 on a Cisco MC3810 multiservice concentrator, enter the **show running-config** command. The trunks do not have to be connected and active. The following is a sample output from this command.

```
Router# show running-config

Building configuration...

Current configuration:
.
.
.
voice class permanent 100
signal timing idle suppress-voice 2000
signal timing oos restart 1000
.
.
.
voice-port 0:0
 voice-class permanent 100
 compand-type a-law
!
voice-port 0:1
 voice-class permanent 100
 compand-type a-law
!
voice-port 0:2
 voice-class permanent 100
 compand-type a-law
.
.
.
```

To display the status of trunk-conditioning signaling and timing parameters for a voice port on a Cisco MC3810 multiservice concentrator, enter one of the following commands:

- **show voice trunk-conditioning signaling**. The following output sample is for voice port 1/5 on a Cisco MC3810 multiservice concentrator:

```
Router# show voice trunk-conditioning signaling 1/5

1/5 :
TX INFO :slow-mode seq#= 25, sig pkt cnt= 42, last-ABCD=0000
hardware-state ACTIVE signal type is NorthamericanCAS
signal path is OPEN
 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
RX INFO :slow-mode, sig pkt cnt= 37
```

```
missing = 0, out of seq = 0, very late = 0
playout depth = 0 (ms), refill count = 1
prev-seq#= 25, last-ABCD=0000
trunk_down_timer = 4212 (ms), idle timer = 0 (sec),
tx_oos_timer = 0 (sec), rx_ais_duration = 0 (ms)
forced playout signal pattern = NONE
signaling playout history
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
```

- **show voice trunk-conditioning signaling summary**. The following output sample is for voice ports on a Cisco MC3810 multiservice concentrator:

```
Router# show voice trunk-conditioning signaling summary

1/1 is shutdown
1/4 is shutdown
1/5 :
TX INFO :slow-mode seq#= 25, sig pkt cnt= 40, last-ABCD=0000
hardware-state ACTIVE signal type is NorthamericanCAS signal path is OPEN
RX INFO :slow-mode, sig pkt cnt= 36, prev-seq#= 25, last-ABCD=0000
```

- **show voice call summary**. The following output sample is for voice port 1/5 on a Cisco MC3810 multiservice concentrator:

```
Router# show voice call summary

PORT      CODEC    VAD VTSP STATE            VPM STATE
========= ======== === ===================== ========================
1/1                                          *shutdown*
1/2       -         -  -                      FXSLS_ONHOOK
1/3       -         -  -                      FXSLS_ONHOOK
1/4                                          *shutdown*
1/5       g729r8    n  S_CONNECT             S_TRUNKED
1/6       -         -  -                      EM_ONHOOK
```

Configuring Trunk Connections

This section covers the following three types of trunk connections:

- PLARs (switched) connections enable the user to make a call without dialing any digits. The router uses the digits that follow the command internally to send the call to a dial peer.
- Trunk and tie-line connections are virtual connections to PBXs and are dedicated until disabled.
- OPXs are off-premise extension connections that are used with the Cisco MC3810 concentrators only.

Configuring PLAR (Switched) Connections

To configure a PLAR connection, enter voice-port configuration mode for the required voice port.

Note The syntax of the **voice-port** command is hardware specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference* for more information.

To configure a PLAR connection, use the following command in voice-port configuration mode:

Command	Purpose
`Router(config-voice-port)# connection plar string`	Specifies a PLAR connection and associates a peer directly with an interface. The *string* argument is a destination telephone number. Valid entries are any series of digits that specify the E.164 standard.

Configuring Trunk/Tie-Line Connections

The following restrictions apply to the trunk/tie-line configuration:

- Trunk/tie-line connections are applicable only to Cisco 2600 and 3600 series routers.
- Use the following voice port combinations:
  - E&M to E&M (same type)
  - FXS to FXO
  - FXS to FXS (without signaling)
- Do not perform number expansion on the destination pattern telephone numbers configured for trunk connection.
- Configure both end routers to establish the trunk connection.
- Use the **shutdown/no shutdown** command sequence on the voice port to activate the configuration.

To configure a trunk or tie-line connection, use the following commands in dial-peer configuration mode for the required POTS dial peer:

	Command	Purpose
Step 1	`Router(config-dial-peer)# destination-pattern [+]string [T]`	Defines the telephone number associated with the POTS dial peer. The keywords and argument are as follows: • Plus sign (**+**)—(Optional) Character indicating an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810. • *string*—Series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – Asterisk (*) and pound sign (#) that appear on standard touch-tone dial pads. On the Cisco 3600 only, these characters cannot be used as leading characters in a string (for example, *650).

	Command	Purpose
		– Comma (,) inserts a pause between digits. – Period (.) matches any entered digit (this character is used as a wildcard). On the Cisco 3600, the period cannot be used as a leading character in a string (for example, .650). • **T**—Indicates that the control character that the destination-pattern value is a variable length dial-string.
Step 2	`Router(config-dial-peer)# port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Associates the POTS dial peer with a specific logical dial interface. The arguments are as follows: • *slot-number*—Location of the voice interface card. Valid entries are from 0 to 3, depending on the slot where the card is installed. • *subunit-number*—Subunit on the voice interface card where the voice port is located. Valid entries are 0 and 1. • *port*—Voice-port number. Valid entries are 0 and 1. • *slot*—Router location of the installed voice port adapter. Valid entries are from 0 to 3. • *port*—Voice interface card location. Valid entries are from 0 to 3. • *ds0-group-no*—Defined DS0 group number. Each group number is represented on a separate voice port. This enables definition of individual DS0s on the digital T1/E1 card.
Step 3	`Router(config-dial-peer)# prefix string`	(Optional) Specifies the prefix for this POTS dial peer. The *string* argument is sent to the telephony interface first, before the telephone number (destination pattern) associated with the dial peer is sent.
Step 4	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode.
Step 5	`Router(config)# dial-peer voice number voip`	Configures a VoIP peer. The *number* argument uniquely identifies the VoIP dial peer.
Step 6	`Router(config-dial-peer)# destination-pattern [+]string [T]`	Defines the destination telephone number associated with this VoIP dial peer.

	Command	Purpose
Step 7	`Router(config-dial-peer)# session target {ipv4:`*destination-address* `| dns:[$s$. | $d$. | $e$. | $u$.]`*host-name* `| loopback:rtp | loopback:compressed | loopback:uncompressed | ras}`	Identifies the IP address of the appropriate port on the destination end router. The keywords and arguments are as follows: • **ipv4:***destination-address*—Specifies the IP address of the dial peer. • **dns:***host-name*—Specifies the domain name server that is the name of the IP address. Valid entries are characters representing the name of the host device. – **s.**—Source destination pattern is part of the domain name. – **d.**—Destination number is part of the domain name. – **e.**—Called number digits are reversed, periods are added in-between each digit of the called number. The string is part of the domain name. – **u.**—Unmatched portion of the destination pattern (such as a defined extension number) is part of the domain name. • **loopback:rtp**—Specifies that all voice data is looped back to the originating source. Applicable for VoIP peers. • **loopback:compressed**—Specifies that all voice data is looped back in compressed mode to the originating source. Applicable for POTS peers. • **loopback:uncompressed**—Specifies that all voice data is looped back in an uncompressed mode to the originating source. Applicable for POTS peers. • **ras**—Indicates that the RAS signaling function protocol is used. A gatekeeper will translate the E.164 address into an IP address.
Step 8	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode and returns to global configuration mode.

	Command	Purpose
Step 9	`Router(config)# voice-port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Enters voice-port configuration mode. The arguments are as follows: • *slot-number*—Defines the location of the voice interface card. Valid entries are from 0 to 3, depending on the slot where the card is installed. • *subunit-number*—Specifies the subunit on the voice interface card where the voice port is located. Valid entries are 0 and 1. • *port*—Specifies the voice-port number. Valid entries are 0 and 1. • *slot*—Defines the router location of the installed voice port adapter. Valid entries are from 0 to 3. • *port*—Indicates the voice interface card location. Valid entries are from 0 to 3. • *ds0-group-no*—Defines the DS0 group number. Each group number is represented on a separate voice port. This enables definition of individual DS0s on the digital T1/E1 card.
Step 10	`Router(config-voice-port)# connection {tie-line | trunk [answer-mode]} string`	Specifies a tie-line connection to a PBX. The keywords and arguments are as follows: • **tie-line**—Used only on the Cisco MC3810 multiservice concentrator when additional prefixed digits are required. The combined set of digits route the call into the network using the dial peers. The tie-line digits are automatically stripped by a terminating port. • **trunk**—Specifies a straight tie-line connection to a PBX. • **answer-mode**—(Optional) Specifies that the router should not attempt to initiate a trunk connection, but should wait for an incoming call before establishing the trunk. If one of the devices is for receiving calls only, use this option. • *string*—Specifies the destination telephone number configured for the destination VoIP dial peer. The value configured for the **connection trunk** command must match the value configured for the VoIP dial peer exactly.

Configuring PLAR-OPX Connections

The **plar-opx** command is specific to the Cisco MC3810 concentrator and configures an OPX connection. The local voice port provides a local response before the remote voice port receives an answer. On FXO interfaces, the voice port does not answer until the remote side answers.

To configure a PLAR-OPX connection, use the following command in voice-port configuration mode for the required voice port:

Command	Purpose
`Router(config-voice-port)# connection plar-opx string`	Specifies a PLAR-OPX connection, associating a peer directly with an interface.

Configuring T1/E1 Alarm Generation Parameters

A network can be configured to monitor any combination of DS0 groups on a T1 or E1 trunk. An alarm is triggered only if all of the monitored DS0 groups on a T1 or E1 trunk are OOS. If one monitored DS0 group is in service, no alarm is triggered. The DS0 groups can be either of the following types:

- DS0 groups configured as voice ports for permanent point-to-point voice connections created using the **connection** command (for private lines and tie-lines). These DS0 groups can go OOS due to a trunk-conditioning event or busyout event (except forced busyout).
- DS0 groups configured as voice ports for switched voice traffic using CAS. These DS0 groups can go OOS, because of a busyout event (except forced busyout).

Note Alarm conditioning is not supported on CCS trunks.

To specify the DS0 group to be monitored and the alarm type, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {t1 | e1} {0 | 1}`	Enters controller configuration mode.
Step 2	`Router(config-controller)# mode {cas | atm}`	Configures the controller for CAS.
Step 3	`Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type {e&m-immediate | e&m-delay | e&m-wink | fxs-ground-start | fxs-loop-start | fxo-ground-start | fxp-loop-start}`	Configures DS0 groups on the controller. The keywords and arguments are as follows: • *ds0-group-no*—Identifies the DS0 group and must be a value from 0 to 23 for T1 and 0 to 30 for E1. • **timeslots** *timeslot-list*—Specifies a single time slot number, a single range of numbers, or multiple ranges of numbers separated by commas. For T1/E1, allowable values are from 1 to 24. Examples are: – 2 – 1–15, 17–24 – 1–23 – 2, 4, 6–12

Command	Purpose
	• **type**—Specifies the signaling method that depends upon the connection. The E&M interface enables connection for PBX lines and telephone equipment. The FXS interface connects basic telephone equipment and the PBX. The FXO interface connects the CO to a standard PBX interface where permitted by local regulations. It is often used for OPXs. The keywords are as follows: – **e&m-immediate** specifies no specific off-hook and on-hook signaling. – **e&m-delay** specifies that the originating endpoint sends an off-hook signal and then and waits for an off-hook signal followed by an on-hook signal from the destination. – **e&m-wink** specifies that the originating endpoint sends an off-hook signal and waits for a wink signal from the destination. – **fxs-ground-start** specifies FXS ground-start signaling support. – **fxs-loop-start** specifies FXS loop-start signaling support. – **fxo-ground-start** specifies FXO ground-start signaling support. – **fxo-loop-start s**pecifies FXO loop-start signaling support. Repeat Step 3 for each DS0 group to be configured.
Step 4 `Router(config-controller)# alarm-trigger blue` *ds0-group-list*	Enables alarm conditioning and configures the system to monitor one or more DS0 groups. The keyword and argument is as follows: • **blue**—Specifies an AIS alarm and is required. • *ds0-group-list*—Values can be a single DS0 group number, a single range of numbers, or multiple ranges of numbers separated by commas. Allowable values are from 0 to 23 for T1 and from 0 to 30 for E1.

Verifying Alarm-Generation Parameters

Use one or more of the following methods to verify that the T1/E1 controller is correctly configured for generating alarms:

- Enter the **show running-config** command. The following output sample is for a Cisco MC3810 multiservice concentrator, with controller E1 0 configured so that a blue alarm is generated if DS0 groups 0, 1, and 2 (voice ports 0:0, 0:1, and 0:2) are all busied out:

```
Router# show running-config

Building configuration...
.
controller E1 0
 mode cas
 ds0-group 0 timeslots 1-10 type e&m-immediate-start
 ds0-group 1 timeslots 11-15,17-20 type e&m-immediate-start
 ds0-group 2 timeslots 21-30 type e&m-immediate-start
 alarm-trigger blue 0-2
```

- Create an OOS state on all voice ports on the controller (this should cause a blue alarm to be generated).
  - For voice ports with the busyout monitor function enabled (switched or trunked), busy out the voice ports by completing the following two steps:
    Step 1. Shut down or disconnect any serial and Ethernet interfaces that are monitored for OOS busyout.
    Step 2. Activate any serial and Ethernet interfaces that are monitored for in-service busyout.

Note All the configured voice ports for switched connections and monitored for alarm trigger must have the busyout monitor function enabled; otherwise, no alarm can be triggered.

  - For voice ports with the busyout monitor function disabled (trunked only), create an OOS condition on the trunks by shutting down or disconnecting the associated local serial interface, or by shutting down the associated far-end T1/E1 controller.
- Enter the **show controller** command. This displays the alarm status of the T1 or E1 trunk on a Cisco MC3810. A yellow alarm is received and detected, and a blue alarm is generated and transmitted:

```
Router# show controller t1 0

T1 0 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  Yellow alarm detected.
  alarm-trigger is set to Blue
  Alarm is triggered
  Slot 3 CSU Serial #00000056 Model TEB HWVersion 3.70 RX level = 0DB
  Framing is ESF, Line Code is B8ZS, Clock Source is Line.
  Data in current interval (827 seconds elapsed):
```

Configuring PSTN Fallback

The following restrictions and limitations apply to PSTN fallback:

- When network congestion is detected, the fallback feature does not affect the existing call. It affects only subsequent calls.
- There can only be one ICPIF/delay-loss value per system.
- There is a small additional call setup delay for the first call to a new IP destination.
- PSTN fallback is supported for H.323 VoIP calls only.

The following sections describe the configuration tasks for PSTN fallback. Each task in the list is identified as either optional or required:

- Configuring Fallback to Alternate Dial Peers, page 592 (required)
- Configuring Destination Monitoring without Fallback to Alternate Dial Peers (optional)
- Configuring Call Fallback Cache Parameters (optional)
- Configuring Call Fallback Jitter-Probe Parameters (optional)
- Configuring Call Fallback Probe-Timeout and Weight Parameters (optional)
- Configuring Call Fallback Threshold Parameters (optional)
- Verifying PSTN Fallback Configuration (optional)

Configuring Fallback to Alternate Dial Peers

To configure fallback to alternate dial peers in case of network congestion, use the following command in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call fallback active`	Enables the PSTN fallback feature to alternate dial peers in case of network congestion.
Step 2	`Router(config)# call fallback key-chain name-of-chain`	Specifies MD5 configuration.

Configuring Destination Monitoring without Fallback to Alternate Dial Peers

To monitor call destinations without fallback to alternate dial peers, use the following command in global configuration mode:

Command	Purpose
`Router(config)# call fallback monitor`	Enables the monitoring of destinations without fallback to alternate dial peers.

Configuring Call Fallback Cache Parameters

To configure the call fallback cache parameters, use the following commands beginning in beginning in configuration mode:

	Command	Purpose
Step 1	`Router(config)# call fallback cache-size number`	Specifies the call fallback cache size.
Step 2	`Router(config)# call fallback cache-timeout seconds`	Specifies the time after which the cache entry is purged. Default is 600 seconds.
Step 3	`Router# clear call fallback cache [ip-address]`	Clears the current ICPIF estimates for all IP addresses or a specific IP address in the cache.

Configuring Call Fallback Jitter-Probe Parameters

To configure the call fallback jitter-probe parameters, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call fallback jitter-probe num-packets number-of-packets`	Specifies the number of packets for jitter. Default is 15 packets.
Step 2	`Router(config)# call fallback jitter-probe precedence precedence`	Specifies the treatment of the jitter-probe transmission. Default is two.
Step 3	`Router(config)# call fallback jitter-probe priority-queue`	Assigns a priority to the queue for jitter probes.

Configuring Call Fallback Probe-Timeout and Weight Parameters

To configure call fallback probe-timeout and weight parameters, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call fallback probe-timeout seconds`	Sets the timeout for an SAA probe. Default is 30 seconds.
Step 2	`Router(config)# call fallback instantaneous-value-weight weight`	Configures the call fallback subsystem to take an average from the last two probes registered in the cache for call requests.

Configuring Call Fallback Threshold Parameters

To configure the call fallback threshold parameters that monitor network traffic for call requests, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# call fallback threshold delay delay-value loss loss-value`	Specifies fallback threshold to use packet delay and loss values. Delay and loss have no default values.
or	
`Router(config)# call fallback threshold icpif threshold-value`	Specifies fallback threshold to use the Calculated Planning Impairment Factor (ICPIF) threshold for network traffic.

Configuring Call Fallback Map Parameters

To configure the call fallback map parameters, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# call fallback map map target ip-address address-list ip-address1 ip-address2 ... ip-address7`	Specifies the call fallback router to keep a cache table (by IP addresses) of distances for several destination peers sitting behind the router.
or	
`Router(config)# call fallback map map target ip-address subnet ip-network netmask`	Specifies the call fallback router to keep a cache table (by subnet addresses) of distances for several destination peers sitting behind the router.

Verifying PSTN Fallback Configuration

To verify the PSTN fallback configuration, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# show call fallback cache`	Displays the current ICPIF estimates for all IP addresses in the call fallback cache.
`Router# show call fallback config`	Displays the current configuration.
`Router# show call fallback stats`	Displays the call fallback statistics.

Troubleshooting Tips

To troubleshoot PSTN fallback, use the following debug commands and ensure that VoIP is working before PSTN fallback is configured:

- **debug call fallback detail** to display details of the VoIP call fallback.
- **debug call fallback probes** to verify that probes are being sent correctly.

Monitoring and Maintaining PSTN Fallback

To monitor and maintain PSTN fallback, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router# clear call fallback cache`	Clears the current ICPIF estimates for all IP addresses in the cache.
`Router# clear call fallback stats`	Clears the call fallback statistics.
`Router# debug call fallback detail`	Displays details of the voice fallback.
`Router# debug call fallback probes`	Displays details of the voice fallback probes.
`Router# test call fallback probe` *ip-address*	Tests a probe to a particular IP address and displays the ICPIF SAA values.

Configuring Local Voice Busyout

This section contains configuration information for the following features:

- Configuring the Busyout Trigger Event, page 596
- Configuring Busyout of Voice Ports, page 596
- Configuring a Voice Port to Monitor the Link to a Remote Interface, page 600
- Configuring a Busyout Monitoring Voice Class, page 601

The following restrictions and limitations apply to the local voice busyout feature:

- A maximum of 32 network interfaces can be monitored for a voice port.
- The busyout feature is not activated when there are no DSP resources or bandwidth available. These two conditions can be addressed by configuring alternate routing.
- This feature is not supported on the BVM.
- When the Cisco MC3810 concentrator is configured, the busyout feature is not activated if there are no DSP resources or bandwidth available. These two conditions can be addressed by configuring alternate routing.

A busyout trigger event can be configured at both the serial interface level and the voice-port level. If there is a conflict between the interface-level trigger event and the voice-port level trigger event (trigger events for each are different), the voice-port-level trigger event overrides the interface level trigger event.

If more than one interface is configured for a busyout trigger event, voice ports are not busied out until all of the interfaces are down.

Note ITU-T G.113, *General Characteristics of International Telephone Connections and Telephone Circuits*, is supported.

Configuring the Busyout Trigger Event

To configure the voice-port busyout trigger event for a serial or ATM network interface, select the required interface and use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# voice-port busyout`	Busies out all voice ports associated with this serial interface. **Note** This command does not busy out any voice ports configured to busy out under specific conditions, as described in the "Forcing the Voice Port into Busyout State" section on page 599.
Step 2	`Router(config-if)# ctrl z`	Exits interface configuration mode and enters EXEC mode.
Step 3	`Router# show voice busyout`	Displays the voice busyout status.

Note When voice-port busyout from a serial network interface is configured, all voice ports are placed into a busyout state if the serial interface goes down.

Configuring Busyout of Voice Ports

A voice port can be configured to busy out under specified conditions or it can be manually forced into a busyout state using the following procedures:

- Configuring Busyout Under Specified Conditions, page 597
- Configuring Seize Conditions, page 598
- Forcing the Voice Port into Busyout State, page 599

The default is to busyout when the monitored interface is OOS.

Configuring Busyout Under Specified Conditions

To configure the busyout of a voice port under specified conditions, use the following command in voice-port configuration mode for the required voice port:

Command	Purpose
`Router(config-voice-port)# busyout monitor interface {serial interface-number | ethernet interface-number} [in-service]`	Specifies an interface to be monitored. When multiple interfaces are configured for OOS, busy out occurs only if all of the interfaces are OOS. When multiple interfaces are configured for in-service, busy out occurs only when any one interface returns to service. The keywords and arguments are as follows: • **serial**—Specifies monitoring of a serial interface. More than one can be entered for a voice port. • *interface-number*—Identifies an interface to be monitored for the voice-port busyout function. • **ethernet**—Specifies monitoring of an ethernet interface. • *interface-number*—Identifies an interface to be monitored for the voice-port busyout function. • **in-service**—Configures the voice port for busy out when the monitored interface returns to service. **Note** The **voice-port** command is hardware specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference* for more information. **Note** Reenter the command for each additional interface to be monitored.

Configuring Seize Conditions

To configure seize conditions, use the following commands in voice-port configuration mode for the required voice port:

	Command	Purpose
Step 1	`Router(config-voice-port)# busyout seize {ignore | repeat}`	For FXO and FXS only. Configures the busyout seize action for this voice port. The keywords are as follows: • **ignore**—Leaves the loop open and ignores the incoming signal. • **repeat**—Seizes the far end and ignores all incoming signals until the far end releases. Remove the seize signal and wait for one second before starting to seize the far end again. **Note** For E&M voice ports, the busyout action is always to seize the far-end line. **Note** The **voice-port** command is hardware specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference* for more information.
Step 2	`Router(config-voice-port)# ctrl z`	Exits voice port configuration mode and enters EXEC mode.
Step 3	`Router# show voice port`	Displays the configured busyout seize actions for the voice ports.

Note The Cisco MC3810 multiservice concentrator returns the voice ports to an idle state when the event that triggered the busyout disappears.

The busyout seize action depends on the voice port signaling type. Table 48 contains information on the busyout actions that take place. For E&M voice ports, the busyout action is always seize.

Table 48 Procedure Settings and Busyout Actions

Voice-Port Signaling Types	Procedure Settings (Busyout-Option Command)	Busyout Actions
FXS Loop Start	Default	Removes the power from the loop. For analog voice ports, this is equivalent to removing the ground from the tip lead. For digital voice ports, the port generates the bit pattern equivalent to removing the ground from the tip lead or busies out if the bit pattern exists.
FXS Loop Start	Ignore	Ignores the ground on the ring lead.
FXS Ground Start	Default	Grounds the tip lead and stays at this state.
FXS Ground Start	Ignore	Leaves the tip lead open and ignores the ground on the ring lead.
FXS Ground Start	Repeat	Grounds the tip lead and waits for the far end to close the loop. The far end closes the loop. If the far end opens the loop again, the FXS removes the ground from the tip lead. FXS waits for several seconds before starting the process again.

Table 48 Procedure Settings and Busyout Actions (continued)

Voice-Port Signaling Types	Procedure Settings (Busyout-Option Command)	Busyout Actions
FXO Loop Start	Default	Closes the loop and stays at this state.
FXO Loop Start	Ignore	Leaves the loop open and ignores the ringing current on the ring level.
FXO Loop Start	Repeat	Closes the loop. After the detected far end starts the power denial procedure, FXO opens the loop. After the detected far end has completed the power denial procedure, FXO waits for several seconds before starting the process again.
FXO Ground Start	Default	Grounds the tip lead.
FXO Ground Start	Ignore	Leaves the loop open and ignores the running current on the ring lead or ground on the tip lead.
FXO Ground Start	Repeat	Grounds the ring lead and removes the ground from the ring lead. Closes the loop after the detected far end grounds the tip lead. When the detected far end removes the ground from the tip lead, FXO opens the loop. The FXO waits for several seconds before starting the process again.
E&M Immediate Start	Default (only option available)	Seizes the far end by setting lead busy.
E&M Delay Start	Default (only option available)	Seizes the far end by setting lead busy.
E&M Wink Start	Default (only option available)	Seizes the far end by setting lead busy.

Forcing the Voice Port into Busyout State

When busyout is configured, the specified voice port is forced into a busyout state when the interface is down. When the **busyout forced** command is entered, the voice port is forced unconditionally into a busyout state. If more than one interface has the **voice-port busyout** interface command configured, all interfaces must be down for the busyout to take effect.

To configure the voice port for a forced busyout state, use the following commands in voice-port configuration mode for the required voice port:

	Command	Purpose
Step 1	`Router(config-voice-port)# busyout forced`	Places the voice port into the busyout state. **Note** If **no busyout forced** is entered, the busyout state is controlled by the **busyout monitor interface** command. If the **busyout monitor interface** command has not been entered, the **no busyout forced** command forces the voice port out of the busyout state. **Note** The **voice-port** command is hardware specific. Refer to the *Cisco IOS Voice, Video, and Fax Command Reference* for more information.
Step 2	`Router(config-voice-port)# ctrl z`	Exits voice-port mode and enters EXEC mode.
Step 3	`router# show voice busyout`	Displays the busyout status.

Note When the voice port is forced into the busyout state, it must be manually forced out of the busyout state by entering the **no busyout forced** command.

Configuring a Voice Port to Monitor the Link to a Remote Interface

The following restrictions and limitations apply to SAA probe monitoring of remote interfaces:

- A maximum of 32 network interfaces can be monitored for a voice port.
- The maximum number of simultaneous SAA probes is controlled by the SAA subsystem design and its configuration.
- Busyout based on monitoring of a remote, IP-addressable interface is not activated when DSP resources and bandwidth are unavailable.
- PSTN Fallback must be enabled for the **busyout monitor probe** command to function.
- PSTN Fallback must also be configured on the router and the SAA responder on the target router.
- The SAA responder function must be enabled on the router at the remote IP address targeted by the SAA probe.
- The SAA probe feature can be configured on CAS trunks only (not CCS).
- If a voice port monitors multiple links, busyout occurs only when *all* of the monitored links go below the threshold.

Individual voice ports can be configured for busyout, or a voice class can be applied that includes all of the busyout parameters (see the "Assigning Voice Classes to Voice Ports" section on page 582).

Note If a busyout voice class has already been assigned to a voice port, a busyout using an SAA probe cannot be configured using this procedure.

To configure a voice port to monitor the link to a remote interface, use the following command in voice-port configuration mode:

Command	Purpose
`Router(config-voice-port)# busyout monitor probe` *ip-address* `[codec` *codec-type*`] [icpif` *number* `| loss` *percent* `delay` *milliseconds*`]`	Configures the busyout probe that monitors the link to the remote interface identified by an IP address. Reenter the command for each additional interface to be monitored. The keywords and arguments are as follows: • *codec-type*—(Optional) Specifies the SAA probe signal. • **icpif** *number*—(Optional) Specifies a threshold for ICPIF. • **loss** *percent* **delay**—(Optional) Specifies a threshold in *milliseconds*, or specifies loss and delay thresholds individually. **Note** If **icpif** values are not entered, the packet delay values from the **call fallback active** command are used.

Verifying the Voice-Port Busyout Configuration

Complete the following tasks to verify that a voice port is correctly configured to monitor the link to a remote interface:

- Shut down the remote interface associated with the configured IP address. This busies out the voice port.
- Enter the **show voice busyout** command to display information about the busyout state. The following is a sample display for voice ports on a Cisco MC3810:

```
Router# show voice busyout

Voice port busyout will be triggered by the
following network interfaces states
 1/1 probe 209.165.202.128 codec g711u icpif 25
 1/2 probe 209.165.202.128 codec g711u icpif 25
 1/3 probe 209.165.202.128 codec g711u icpif 25
The following voice ports are in busyout state

1/1is in busyout state caused by
probe 209.165.202.128 codec g711u icpif 2
1/2is in busyout state caused by
probe 209.165.202.128 codec g711u icpif 2
1/3is in busyout state caused by
probe 209.165.202.128 codec g711u icpif 2
```

Configuring a Busyout Monitoring Voice Class

A busyout voice class monitors local ports (serial and Ethernet) and links to remote IP addresses. Busyout occurs when all of the monitored local ports are OOS *or* when all of the monitored links go below the configured threshold value. If a voice port is configured to monitor multiple links, busyout occurs only when *all* of the monitored links go below the threshold.

To define a voice class with specified busyout conditions, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class busyout` *tag*	Creates a voice class for defining busyout conditions. The range for the *tag* number is 1 to 10000. The *tag* number must be unique on the router.
Step 2	`Router(config-voice-class)# busyout monitor serial` *interface-number* [**in-service**]	(Optional) Specifies a local serial interface to be monitored by the voice port. To configure the voice port to monitor multiple interfaces, reenter the command for each additional interface to be monitored.[1]

	Command	Purpose
Step 3	`Router(config-voice-class)# busyout monitor ethernet` *interface-number*} [**in-service**]	(Optional) Specifies a local Ethernet interface to be monitored by the voice port. To configure the voice port to monitor multiple interfaces, reenter the command for each additional interface.[1]
Step 4	`Router(config-voice-class)# busyout monitor probe` *ip-address* [**codec** *codec-type*] [**icpif** *number* | **loss** *loss-value* **delay** *milliseconds*]	(Optional) Configures the voice port to use an SAA probe to monitor the link to the remote interface identified by an IP address. (Optional) Specifies a codec profile for the SAA probe signal and ICPIF loss/delay threshold or loss and delay thresholds individually. Packet loss and delay determine the threshold for initiating the busyout state. **Note** To configure the voice port to monitor multiple remote interfaces, reenter the command for each additional interface to be monitored. If a threshold value is not entered, the packet delay values from the **call fallback active** command are used. **Note** PSTN fallback must be configured on this router and the SAA responder on the target router.

1. The default is that the voice port is busied out when the monitored interface is OOS. Enter the keyword **in-service** to configure the voice port for busyout when the monitored interface comes into service. If a voice port is configured to monitor multiple interfaces for OOS, busyout occurs only when *all* the monitored serial and Ethernet interfaces are OOS. If a voice port is configured to monitor multiple interfaces for the in-service state, busyout occurs when *any one* monitored serial or Ethernet interface comes into service.

After the voice class for the busyout function has been created, assign it to all voice ports that have these busyout requirements. See the "Assigning Voice Classes to Voice Ports" section on page 582.

Verifying the Voice- and Voice-Class Busyout Configuration

To verify the voice-class busyout, assign the voice class to a voice port as described in the "Assigning Voice Classes to Voice Ports" section on page 582, and verify the busyout function of the voice port.

To verify that a voice port is correctly configured for busyout monitoring, perform the following tasks:

- Shut down or bring up the monitored interface or interfaces, as required. The voice port is busied out. Monitored interfaces can be any of the following, depending on the configuration:
  - Local interfaces—for **busyout monitor serial** and for **busyout monitor ethernet**. If the voice port is configured to monitor multiple local interfaces for OOS, busyout occurs only when *all* the monitored interfaces are OOS. If a voice port is configured to monitor multiple local interfaces for the *in-service* state, busyout occurs when *any one* monitored interface comes into service.
  - Remote interface—for **busyout monitor probe**

  The voice port monitors a remote IP address for OOS only.

Note Ensure that PSTN fallback is configured on the local router and SAA responder is configured on the target router.

- Enter the **show voice busyout** command to display information about the busyout state. The following is a sample display for voice ports on a Cisco MC3810 multiservice concentrator:

```
Router# show voice busyout

Voice port busyout will be triggered by the
following network interfaces states
 1/2 busyout monitor ATM0
 1/3 busyout monitor ATM0
 1/4 busyout monitor Serial0
 1/5 busyout monitor Serial0
 1/6 probe 209.165.202.128 codec g711u icpif 25
 The following voice ports are in busyout state

 1/1 is forced into busyout state
 1/2 is in busyout state caused by ATM0
 1/3 is in busyout state caused by ATM0
 1/4 is in busyout state caused by Serial0
 1/5 is in busyout state caused by Serial0
 1/6 is in busyout state caused by probe 209.165.202.128 codec g711u icpif 2
```

Trunk Connections and Conditioning Configuration Examples

This section has the following examples:

Trunk Conditioning Configuration Example

The following example configures a voice class and then applies it to a VoFR and VoATM dial peer on Cisco MC3810 series concentrators:

```
Router(config)# voice class permanent 10
Router(config-class)# signal keepalive 10
Router(config-class)# signal pattern idle receive 0101
Router(config-class)# signal pattern idle transmit 0101
Router(config-class)# signal timing idle suppress-voice 5
Router(config-class)# signal pattern oos receive 0001
Router(config-class)# signal pattern oos transmit 0001
Router(config-class)# signal timing oos timeout 60
Router(config-class)# signal timing oos restart 120
Router(config-class)# signal timing oos suppress-voice 30
Router(config)# dial peer voice vofr 10
```

```
Router(config-dial-peer)# voice-class permanent 10
Router(config)# dial peer voice voatm 20
Router(config-dial-peer)# voice-class permanent 10
```

Voice Class for VoFR and VoATM Dial Peers Configuration Example

The following example configures a voice class using default idle and OOS signaling patterns and configures busyout to the PBX after a 60-second loss of signaling packets, with restart after 120 seconds:

```
Router(config)# voice class permanent 10
Router(config-class)# signal keepalive 10
Router(config-class)# signal timing oos timeout 60
Router(config-class)# signal timing idle suppress-voice 5
Router(config-class)# signal timing oos restart 120
Router(config-class)# exit
Router(config)# dial peer voice vofr 10
Router(config-dial-peer)# voice-class permanent 10
Router(config-dial-peer)# exit
Router(config)# dial peer voice voatm 20
Router(config-dial-peer)# voice-class permanent 10
Router(config-dial-peer)# exit
```

Voice Class for Voice Ports Configuration Example

The following configuration example shows a voice class with specified signaling bit patterns for the idle receive and transmit; OOS receive and transmit states; and busyout to the PBX after a 90-second loss of signaling packets with restart after 240 seconds:

```
Router(config)# voice class permanent 30
Router(config-class)# signal keepalive 10
Router(config-class)# signal pattern idle receive 0101
Router(config-class)# signal pattern idle transmit 0101
Router(config-class)# signal pattern oos receive 0001
Router(config-class)# signal pattern oos transmit 0001
Router(config-class)# signal timing oos timeout 90
Router(config-class)# signal timing idle suppress-voice 5
Router(config-class)# signal timing oos restart 240
Router(config-class)# exit
Router(config)# voice-port 0/1:5
Router(config-voiceport)# voice-class permanent 30
```

Voice Class for Default Signaling Patterns Configuration Example

The following configuration example shows a voice class using default idle and OOS signaling patterns and configures busyout after 60 seconds to the PBX, with restart after 120 seconds. It applies the voice class to both VoFR and VoATM dial peers:

```
Router(config)# voice class permanent 10
Router(config-class)# signal keepalive 10
Router(config-class)# signal timing oos timeout 60
Router(config-class)# signal timing idle suppress-voice 5
Router(config-class)# signal timing oos restart 120
Router(config-class)# exit
Router(config)# dial peer voice vofr 10
Router(config-dial-peer)# voice-class permanent 10
```

```
Router(config-dial-peer)# exit
Router(config)# dial peer voice voatm 20
Router(config-dial-peer)# voice-class permanent 10
Router(config-dial-peer)# exit
```

Voice Class for Specified Signaling Patterns Configuration Example

The following example configures a voice class with specified signaling bit patterns for the idle receive, idle transmit, OOS receive, and OOS transmit states, and it configures busyout after 90 seconds to the PBX, with restart after 240 seconds. It applies the voice class to digital voice port 0:5 on a Cisco MC3810:

```
Router(config)# voice class permanent 30
Router(config-class)# signal keepalive 10
Router(config-class)# signal pattern idle receive 0101
Router(config-class)# signal pattern idle transmit 0101
Router(config-class)# signal pattern oos receive 0001
Router(config-class)# signal pattern oos transmit 0001
Router(config-class)# signal timing oos timeout 90
Router(config-class)# signal timing idle suppress-voice 5
Router(config-class)# signal timing oos restart 240
Router(config-class)# exit
Router(config)# voice-port 0:5
Router(config-voiceport)# voice-class permanent 30
```

PLAR (Switched Calls) Configuration Example

The following example configures the DTMF relay and PLAR for router alpha:

```
hostname router-alpha
!
voice-card 1
!
controller T1 1/0
 framing esf
 linecode b8zs
 ds0-group 1 timeslot 1 type fxo-loop
 ds0-group 2 timeslot 2 type fxo-loop
!
dial-peer voice 1 voip
 dtmf-relay  h245-alpha
 codec g729a
 destination-pattern 2..
 session target ipv4:192.168.100.2
!
dial-peer voice 2 pots
 destination-pattern 101
 port 1/0:1
!
dial-peer voice 3 pots
 destination-pattern 102
 port 1/0:2
!
voice-port 1/0:1
 connection plar 201
!
voice-port 1/0:2
 connection plar 202
!
```

```
interface s0/0
 ip address 192.168.100.1 255.255.255.0
```

The following example configures the DTMF relay for router beta:

```
hostname router-beta
!
dial-peer voice 1 voip
 destination-pattern 1..
 dtmf-relay h245-alpha
 codec g729a
 session target ipv4:192.168.100.1
!
dial-peer voice 2 pots
 destination-pattern 201
 port 1/1
!
dial-peer voice 3 pots
 destination-pattern 202
 port 1/2
!
voice-port 1/1
!
voice-port 1 / 2
!
interface serial 0/0
 ip address 192.168.100.2 255.255.255.0
```

Permanent Trunks Configuration Example

A trunk connection can be used only between E&M ports or with FXO-to-FXS connections. The following example configures the alpha router:

```
hostname router-alpha
!
voice-card 1
!
controller T1 1/0
 framing esf
 linecode b8zs
 ds0-group 1 timeslot 1 type e&m-wink
 ds0-group 2 timeslot 2 type e&m-wink
 clock source line
!
voice-port 1/0:1
 connection trunk 1111
!
voice-port 1/0:2
 connection trunk 1112
!
dial-peer voice 1 voip
 dtmf-relay h245-alpha
 codec g729a
 destination-pattern 111.
 session target ipv4:192.168.100.2
!
dial-peer voice 2 pots
 destination-pattern 2221
port 1/0:1
!
```

```
dial-peer voice 3 pots
 destination-pattern 2222
 port 1/0:2
!
interface serial 0/0
 ip address 192.168.100.1 255.255.255.0
```

The following example configures the beta router:

```
hostname router-beta
!
voice-card 1
!
controller T1 1/0
 framing esf
 linecode b8zs
 ds0-group 1 timeslot 1 type e&m-wink
 ds0-group 2 timeslot 2 type e&m-wink
 clock source line
!
voice-port 1/0:1
 connection trunk 2221
!
voice-port 1/0:2
 connection trunk 2222
!
dial-peer voice 1 voip
 dtmf-relay h245-alpha
 codec g729a
 destination-pattern 222.
 session target ipv4:192.168.100.1
!
dial-peer voice 2 pots
 destination-pattern 1111
 port 1/0:1
!
dial-peer voice 3 pots
 destination-pattern 1112
 port 1/0:2
!
interface serial 0/0
 ip address 192.168.100.2 255.255.255.0
```

In this configuration, a permanent and transparent path is set up between individual DS0s on each router. It passes dial tone from the remote PBX and passes DTMF digits out of band.

The **connection** command, using the keyword **trunk**, establishes the permanent trunk connection between the routers. The digits after the command are passed internally within the router to match a dial peer so that the call can be set up.

Congestion Monitoring and Management Configuration Examples

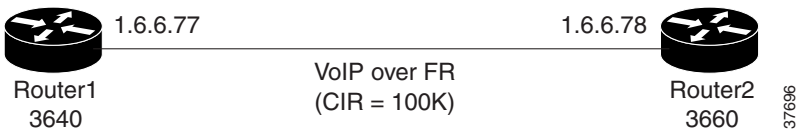
This section has the following examples:

Configuring PSTN Fallback for VoIP over Frame Relay Example

The following output sample shows the PSTN fallback configuration with default fallback values on Router1 for VoIP over Frame Relay as shown in Figure 123. The direction of the calls is from Router1, a Cisco 3640, to Router2, a Cisco 3660. In this example, MD5 authentication is not configured.

Also, SAA responder is configured on Router2 to answer the probes from Router1. When the number 3666 is called from Router1 and congestion is on the link between 10.6.6.77 and 10.6.6.78, the call is not admitted. The user hears a busy tone because there is only one dial peer, 3666, and the IP network that is connected to it is congested. To help avoid this congestion, the **call fallback active** command is enabled here for PSTN fallback. No other call fallback parameters have been configured.

Figure 123 Network Example for VoIP over Frame Relay




```
Router(config)# show running-config
Current configuration:
!
version 12.2
service timestamps debug datetime msec localtime
service timestamps log uptime
no service password-encryption
!
hostname Router1
!
voice-card 3
!
ip subnet-zero
no ip domain-lookup
!
frame-relay switching
!
call fallback active
!
interface Ethernet0/0
 ip address 10.3.3.77 255.255.0.0
 no ip directed-broadcast
!
interface Serial0/0
 no ip address
```

```
 no ip directed-broadcast
 encapsulation frame-relay
 load-interval 30
 no keepalive
 frame-relay traffic-shaping
 frame-relay inverse-arp interval 15
!
interface Serial0/0.1 point-to-point
 ip address 10.6.6.77 255.255.0.0
 no ip directed-broadcast
 frame-relay interface-dlci 100
  class frs0
!
interface Ethernet0/1
 ip address 10.4.4.77 255.255.0.0
 no ip directed-broadcast
 load-interval 30
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
ip route 10.5.0.0 255.255.0.0 10.4.4.78
ip route 10.255.254.254 255.255.255.255 Ethernet0/0
no ip http server
!
map-class frame-relay frs0
 no frame-relay adaptive-shaping
 frame-relay cir 100000
 frame-relay bc 560
 frame-relay mincir 100000
 frame-relay fair-queue
 frame-relay fragment 100
 frame-relay ip rtp priority 16384 16383 75
!
line con 0
 exec-timeout 35791 0
 transport input none
line aux 0
line vty 0 4
 password ard
 login
!
voice-port 1/0/0
!
voice-port 1/0/1
!
voice-port 1/1/0
!
voice-port 1/1/1
!
dial-peer voice 10 pots
 destination-pattern 6666
 port 1/0/0
!
dial-peer voice 20 pots
 destination-pattern 6777
 port 1/0/1
!
dial-peer voice 300 voip
 destination-pattern 3...
 no vad
 session target ipv4:10.6.6.78
!
dial-peer voice 60 pots
 destination-pattern 6111
```

```
 port 1/1/0
!
end
```

Call fallback is not configured on this router. Router2 is a dial peer for Router1, but is not handling calls directly from the PSTN. SAA is configured on Router2 to answer the probes from Router1.

```
Router(config)# show running-config

version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router2
!
voice-card 4
!
ip subnet-zero
!
isdn voice-call-failure 0
!
interface FastEthernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
interface Ethernet1/0
 ip address 10.3.22.80 255.255.0.0
 no ip directed-broadcast
!
interface Serial1/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 load-interval 30
 no keepalive
 clockrate 256000
 frame-relay traffic-shaping
 frame-relay inverse-arp interval 15
!
interface Serial1/0.1 point-to-point
 ip address 10.6.6.78 255.255.0.0
 no ip directed-broadcast
 frame-relay interface-dlci 100
  class frs0
!
interface Ethernet1/1
 ip address 10.5.5.74 255.255.0.0
 no ip directed-broadcast
!
map-class frame-relay frs0
 frame-relay fragment 100
 frame-relay ip rtp priority 16384 16383 75
 no frame-relay adaptive-shaping
```

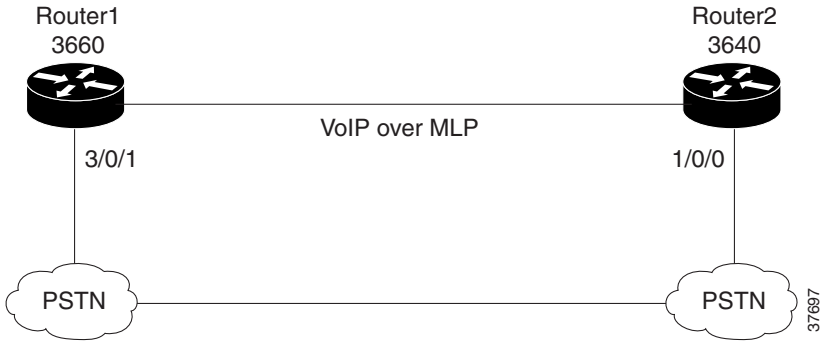
```
 frame-relay cir 100000
 frame-relay bc 1000
 frame-relay mincir 100000
 frame-relay fair-queue
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
!
dial-peer voice 10 pots
 destination-pattern 3111
 port 2/0/0
!
dial-peer voice 20 pots
 destination-pattern 3222
 port 2/0/1
!
dial-peer voice 100 voip
 destination-pattern 6...
 no vad
 session target ipv4:10.6.6.77
!
dial-peer voice 60 pots
 destination-pattern 3999
 port 3/0/0
!
dial-peer voice 70 pots
destination-pattern 3888
 port 3/0/1
!
saa responder
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Configuring PSTN Fallback for VoIP over MLP Example

The following output sample configures PSTN fallback for VoIP over MLP for Router1 shown in Figure 124. The direction of the calls is from Router1, a Cisco 3660, to Router2, a Cisco 3640. MD5 authentication is configured. Also, SAA is configured on Router2 to answer the probes from Router1. When the number 6666 is called from Router1 and congestion is on the link between Router1 and Router2, the call is sent to port 3/0/1 and hence to Router2 over the PSTN.

Probes are sent every 20 seconds (default) with 15 packets in each probe, and are sent in the priority queue with the other voice packets after the **ip rtp priority** command is enabled. Also, the delay and loss threshold command is configured with a delay threshold of 150 milliseconds and a loss threshold of 5 percent, and the cache-aging timeout is 10,000 seconds. The link is configured for 128 kilobits per second (kbps), and 80 kbps is reserved for voice using the **ip rtp priority** command.

Figure 124 Network Example for VoIP over MLP




```
Router (config)# show running-config

Current configuration:
!
version 12.2
service timestamps debug datetime
service timestamps log datetime
no service password-encryption
!
hostname Router1
!
voice-card 4
!
ip subnet-zero
!
call fallback probe-timeout 20
call fallback threshold delay 150 loss 5
call fallback jitter-probe num-packets 15
call fallback jitter-probe priority-queue
call fallback cache-timeout 10000
call fallback active
!
interface Multilink1
 ip address 10.10.10.1 255.255.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no keepalive
 fair-queue 64 256 0
 no cdp enable
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 multilink-group 1
 ip rtp priority 16384 16383 80
!
interface FastEthernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
```

```
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
interface Ethernet1/0
 ip address 10.3.22.80 255.255.0.0
 no ip directed-broadcast
!
interface Serial1/0
 bandwidth 128
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 no fair-queue
 clockrate 125000
 ppp authentication chap
 ppp multilink
 multilink-group 1
!
interface Ethernet1/1
 ip address 10.5.5.74 255.255.0.0
 no ip directed-broadcast
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet1/0
ip route 10.4.0.0 255.255.0.0 10.5.5.78
ip route 10.255.254.254 255.255.255.255 10.3.0.1
no ip http server
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
!
dial-peer voice 10 pots
 destination-pattern 3111
 port 2/0/0
!
dial-peer voice 20 pots
 destination-pattern 3222
 port 2/0/1
!
dial-peer voice 60 pots
 destination-pattern 3999
```

```
 port 3/0/0
!
dial-peer voice 70 pots
 destination-pattern 6666
 port 3/0/1
!
dial-peer voice 200 voip
 destination-pattern 6...
 session target ipv4:10.10.10.1
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 exec-timeout 0 0
 login
!
end
```

SAA is configured on Router2 to answer the probes from Router1:

```
Router (config)# show running-config

version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router2
!
voice-card 4
!
ip subnet-zero
!
isdn voice-call-failure 0
!
interface FastEthernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet0/1
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto
!
interface Ethernet1/0
 ip address 10.3.22.80 255.255.0.0
 no ip directed-broadcast
!
interface Serial1/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 load-interval 30
 no keepalive
 clockrate 256000
 frame-relay traffic-shaping
 frame-relay inverse-arp interval 15
```

```
!
interface Serial1/0.1 point-to-point
 ip address 10.6.6.78 255.255.0.0
 no ip directed-broadcast
 frame-relay interface-dlci 100
  class frs0
!
interface Ethernet1/1
 ip address 10.5.5.74 255.255.0.0
 no ip directed-broadcast
!
map-class frame-relay frs0
 frame-relay fragment 100
 frame-relay ip rtp priority 16384 16383 75
 no frame-relay adaptive-shaping
 frame-relay cir 100000
 frame-relay bc 1000
 frame-relay mincir 100000
 frame-relay fair-queue
!
voice-port 2/0/0
!
voice-port 2/0/1
!
voice-port 2/1/0
!
voice-port 2/1/1
!
voice-port 3/0/0
!
voice-port 3/0/1
!
voice-port 3/1/0
!
voice-port 3/1/1
!
dial-peer voice 10 pots
 destination-pattern 3111
 port 2/0/0
!
dial-peer voice 20 pots
 destination-pattern 3222
 port 2/0/1
!
dial-peer voice 100 voip
 destination-pattern 6...
 no vad
 session target ipv4:10.6.6.77
!
dial-peer voice 60 pots
 destination-pattern 3999
 port 3/0/0
!
dial-peer voice 70 pots
destination-pattern 3888
 port 3/0/1
!
saa responder
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
```

Local Voice Busyout Configuration Examples

The following example configures digital voice port 0:0.4 on a Cisco MC3810 series concentrator to go into the busyout state if serial interface 0:0 goes out of service:

```
Router(config)# voice-port 0:0.4

  Type of VoicePort is FXS
router(config-voiceport)# busyout monitor interface serial 0:0
1/2 is in busyout state

Router(config-voiceport)# end
Router# show voice busyout

!If following network interfaces are down, voice port will be put into busyout state
The following voice ports are in busyout state

1/1 is forced into busyout state
1/2 is in busyout state caused by Serial0
```

The following example configures digital voice port 2/1:7 on a Cisco 3600 series router to go into the busyout state if serial interface 0:0 goes out of service:

```
Router(config)# voice-port 2/1:7

  Type of VoicePort is FXS

Router(config-voiceport)# busyout monitor interface serial 0:0

1/2 is in busyout state

Router(config-voiceport)# end
Router# show voice busyout

!If following network interfaces are down, voice port will be put into busyout state
The following voice ports are in busyout state

2/1:7 is forced into busyout state
2/1:8 is in busyout state caused by Serial0
```

The following example configures the busyout seize action for analog voice port 0/2/1 on a Cisco 3600 series router to repeat:

```
Router(config)# voice-port 0/2/1

  Type of VoicePort is FXO

Router(config-voiceport)# busyout seize repeat
Router(config-voiceport)# end
Router# show voice busyout

!If following network interfaces are down, voice port will be put into busyout state
The following voice ports are in busyout state

0/2/1 is forced into busyout state
0/2/2 is in busyout state caused by Serial0
```

The following example forces DS0 timeslots 1 through 12 on controller T1 0 on a Cisco MC3810 multiservice concentrator into the busyout state:

```
Router(config)# controller t1 0
Router(config-controller)# ds0 busyout 1-12
Router(config-controller)# end
```

The following example configures busyout voice class 35, which initiates voice-port busyout whenever either serial port 0 or 1 is in service, and it applies voice class 35 to voice port 1/3:

```
Router(config)# voice class busyout 35
Router(config-voice-class)# busyout monitor serial 0 in-service
Router(config-voice-class)# busyout monitor serial 1 in-service
Router(config-voice-class)# exit
Router(config)# voice-port 1/3
Router(config-voiceport)# voice class 35
```

The following example configures busyout voice class 40, which initiates voice-port busyout whenever an SAA probe sent to both of the two specified remote interfaces results in a link with an ICPIF delay/loss average of more than 15, and it applies voice class 40 to voice port 1/4:

```
Router(config)# voice class busyout 40
Router(config-voice-class)# busyout monitor probe 209.165.202.128 icpif 15
Router(config-voice-class)# busyout monitor probe 209.165.202.129 icpif 15
Router(config-voice-class)# exit
Router(config)# voice-port 1/4
Router(config-voiceport)# voice class 40
```

The following example configures analog voice port 1/1 on a Cisco MC3810 to use an SAA probe with a G.711 alaw profile to probe the link to the remote interface with IP address 209.165.202.128, and to busyout the voice port if the link has a packet loss of more than 50 percent and a packet delay of more than 25 milliseconds:

```
Router(config)# voice-port 1/1
Router(config-voiceport)# busyout monitor probe 209.165.202.128 codec g711a loss 50
delay 25
```

The following example configures voice port 1/0/1 on a Cisco 3600 series router to use an SAA probe with the default (G.711 ulaw) profile to probe the link to the remote interface with IP address 209.165.202.128, and to busyout the voice port if the link has packet loss and delay that exceed the threshold values configured by the **call fallback active** command:

```
Router(config)# voice-port 1/0/1
Router(config-voiceport)# busyout monitor probe 209.165.202.128
```

The following example configures busyout voice class 60, which configures multiple parameters for voice-port busyout, and it applies voice class 60 to voice ports 1/0/0 and 1/0/1 on a Cisco 3600 series router. The voice ports will busy out under any one the following conditions:

- Serial ports 0/0 and 0/1 are both OOS
- Serial port 1/0 or 1/0 is in service
- The link loss exceeds 50 percent or the link delay exceeds 1 second on the links to both remote interfaces (IP addresses 209.165.202.128 and 209.165.202.129)

```
Router(config)# voice class busyout 60
Router(config-voice-class)# busyout monitor serial 0/0
Router(config-voice-class)# busyout monitor serial 0/1
Router(config-voice-class)# busyout monitor serial 1/0 in-service
Router(config-voice-class)# busyout monitor serial 1/1 in-service
Router(config-voice-class)# busyout monitor probe 209.165.202.128 loss 50 delay 1000
Router(config-voice-class)# busyout monitor probe 209.165.202.129 loss 50 delay 1000
Router(config-voice-class)# exit
```

```
Router(config)# voice-port 1/0/0
Router(config-voiceport)# voice class 60
Router(config-voiceport)# exit
Router(config)# voice-port 1/0/1
Router(config-voiceport)# voice class 60
Router(config-voiceport)# exit
```

The following example configures voice port 1/1 into forced busyout state:

```
Router(config)# voice-port 1/1

Type of VoicePort is FXS

Router(config-voiceport)# busyout forced
00:09:46: port 0 is forced into busyout state

Router(config-voiceport)# end
Router# show voice busyout

!If following network interfaces are down, voice port will be put into busyout state.
The following voice ports are in busyout state
1/1 is forced into busyout state
```

The following example configures voice port 1/2 to busyout monitor mode, monitoring serial 0:

```
Router(config)# voice-port 1/2
Type of VoicePort is FXS

Router(config-voiceport)# busyout-monitor serial 0
1/2 is in busyout state

Router(config-voiceport)# end
Router# show voice busyout

!If following network interfaces are down, voice port will be put into busyout state.
The following voice ports are in busyout state
1/1 is forced into busyout state
1/2 is in busyout state caused by Serial0
```

The following example configures voice port 1/3 to the busyout seize repeat state:

```
Router(config)# voice-port 1/3
Type of VoicePort is FXO

router(config-voiceport)# busyout-seize repeat
Router(config-voiceport)# end
Router# show voice busyout

!If following network interfaces are down, voice port will be put into busyout state.
The following voice ports are in busyout state
1/1 is forced into busyout state
1/2 is in busyout state caused by Serial0
```

Alarm Trigger for Busyout of Voice Ports Configuration Example

This example creates three permanent trunks on controller T1 0 and configures T1 0 to send a blue (AIS) alarm if all three permanent trunks are OOS. These steps create the voice ports and configure the alarm trigger:

```
Router(config)# controller t1 0
Router(config-controller)# mode cas
Router(config-controller)# ds0-group 0 timeslots 1-10 type fxs-ground-start
Router(config-controller)# ds0-group 1 timeslots 11 type fxs-ground-start
Router(config-controller)# ds0-group 2 timeslots 12-23 type fxs-ground-start
Router(config-controller)# alarm-trigger blue 0-2
Router(config-controller)# exit
Router(config)#
```

These steps create a voice class to define the trunk conditioning parameters for permanent trunks (in which the default values are not used):

```
Router(config)# voice class permanent 8
Router(config-class)# signal keepalive 10
Router(config-class)# signal timing oos timeout 60
Router(config-class)# signal timing idle suppress-voice 5
Router(config-class)# signal timing oos restart 120
Router(config-class)# exit
Router(config)#
```

These steps create a VoIP dial peer to define the network connectivity and trunk conditioning parameters for permanent trunks:

```
Router(config)# dial-peer voice 100 voip
Router(config-dial-peer)# session target ipv4:172.20.10.10
Router(config-dial-peer)# destination-pattern 10..
Router(config-dial-peer)# voice-class permanent 8
Router(config-dial-peer)# exit
Router(config)#
```

These steps assign each voice port to a permanent trunk and associate each trunk with a network dial peer:

```
Router(config)# voice-port 0:0
Router(config-voiceport)# connection trunk 1001
Router(config-voiceport)# exit
Router(config)# voice-port 0:1
Router(config-voiceport)# connection trunk 1002
Router(config-voiceport)# exit
Router(config)# voice-port 0:2
Router(config-voiceport)# connection trunk 1003
Router(config-voiceport)# exit
Router(config)#
```

This example configures voice port 0:0 for busyout if serial port 0.1, 0.2, and Ethernet port 0 all go out of service, or serial port 1 comes into service:

```
Router(config)# voice-port 0:0
Router(config-voiceport)# busyout monitor serial 0.1
Router(config-voiceport)# busyout monitor serial 0.2
Router(config-voiceport)# busyout monitor ethernet 0
Router(config-voiceport)# busyout monitor serial 1 in-service
Router(config-voiceport)# exit
```

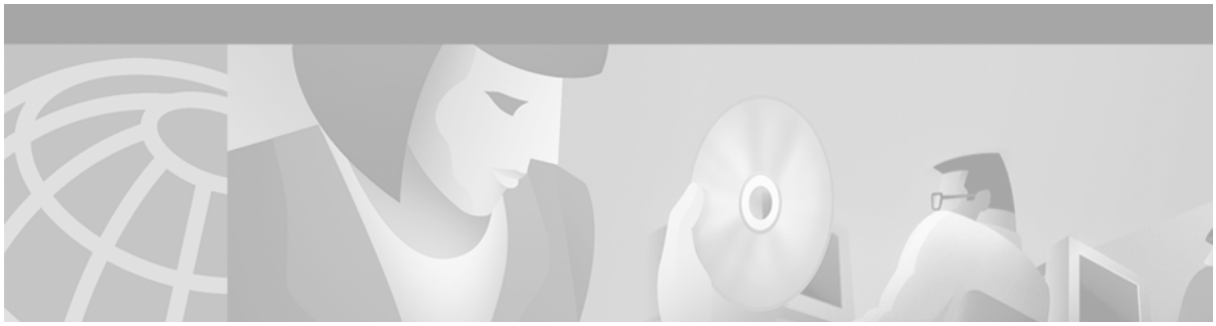
This example configures voice port 0:1 for busyout if the connections to both of two remote IP addresses are OOS:

```
Router(config)# voice-port 0:1
Router(config-voiceport)# busyout monitor probe 209.165.202.128 codec g711a icpif 15
Router(config-voiceport)# busyout monitor probe 209.165.202.129 codec g711a icpif 15
Router(config-voiceport)# exit
```

This example configures voice port 0:2 for busyout under any one of the following conditions:

- Serial port 0.1 and 0.2 are both OOS
- Serial port 1 comes into service
- Connections to both of two remote IP addresses are OOS

```
Router(config)# voice-port 0:2
Router(config-voiceport)# busyout monitor serial 0.1
Router(config-voiceport)# busyout monitor serial 0.2
Router(config-voiceport)# busyout monitor serial 1 in-service
Router(config-voiceport)# busyout monitor probe 209.165.202.128 codec g711a icpif 15
Router(config-voiceport)# busyout monitor probe 209.165.202.129 codec g711a icpif 15
Router(config-voiceport)# exit
Router(config)# exit
```



Configuring ISDN Interfaces for Voice

This chapter explains how to configure ISDN Basic Rate Interface (BRI) and Primary Rate Interface (PRI) ports for voice support and contains the following sections:

For a complete description of the commands used to configure ISDN interfaces for voice, refer to the *Cisco IOS Dial Technologies Command Reference* and the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

The following Cisco devices provide ISDN interfaces for voice applications:

- Cisco 2600 series routers (ISDN BRI and PRI interfaces)
- Cisco 3600 series routers (ISDN BRI and PRI interfaces)
- Cisco 7200 series routers (ISDN PRI interfaces only)
- Cisco MC3810 multiservice concentrator (ISDN BRI interfaces only)
- Cisco AS5300 universal access servers (ISDN PRI interfaces only)
- Cisco AS5800 universal access servers (ISDN PRI interfaces only)

The following documents provide additional information to help implement ISDN interfaces for voice:

- *Cisco IOS IP Configuration Guide*
- *Cisco IOS IP Command Reference*
- *Cisco IOS Dial Technologies Configuration Guide*
- *Cisco IOS Dial Technologies Command Reference*
- *Cisco IOS IOS Voice, Video, and Fax Command Reference*
- *Voice Network Module and Voice Interface Card Configuration Note*
- *Cisco Network Module Hardware Installation Guide*
- *Cisco WAN Interface Cards Hardware Installation Guide*

- *Update to Cisco WAN Interface Cards Hardware Installation Guide*
- *Voice over IP for the Cisco 3600 and Cisco 2600 Series Software Configuration Guide*
- *Cisco 7200 Series Port Adapter Hardware Configuration Guidelines*
- *Cisco 7200 Series Configuration Notes*
- *Quick Start Guide: Cisco MC3810 Installation and Startup*
- *Cisco MC3810 Multiservice Concentrator Hardware Installation Guide*

The following documents can help you troubleshoot ISDN, PRI, and BRI connections:

- *Internetwork Troubleshooting Guide*
- *Cisco IOS Debug Command Reference*

ISDN Voice Interface Overview

ISDN voice support provides the following benefits:

- It allows you to bypass Public Switched Telephone Network (PSTN) tariffed services such as trunking and administration.
- It allows your PBXs to be connected directly to a Cisco router so PBX station calls can be routed automatically to the WAN.
- It allows you to configure a voice interface on a Cisco router to emulate either a Terminating Equipment (TE) or Network Termination (NT) interface. Customers with all types of PBXs can send calls through a Cisco router and deliver those calls across the customer network.
- It allows you to configure Layer 2 operation as point-to-point (static terminal endpoint identifier [TEI]) or point-to-multipoint (automatic TEI).

Cisco routing devices support ISDN BRI and ISDN PRI. Both media types use bearer (B) channels and data (D) channels.

ISDN BRI provides two B channels, each capable of transferring voice or data at 64 kbps, and one 16-kbps D channel that carries signaling traffic. The D channel is used by the telephone network to carry instructions about how to handle each of the B channels. ISDN BRI (also referred to as “2 B + D”) provides a maximum transmission speed of 128 kbps.

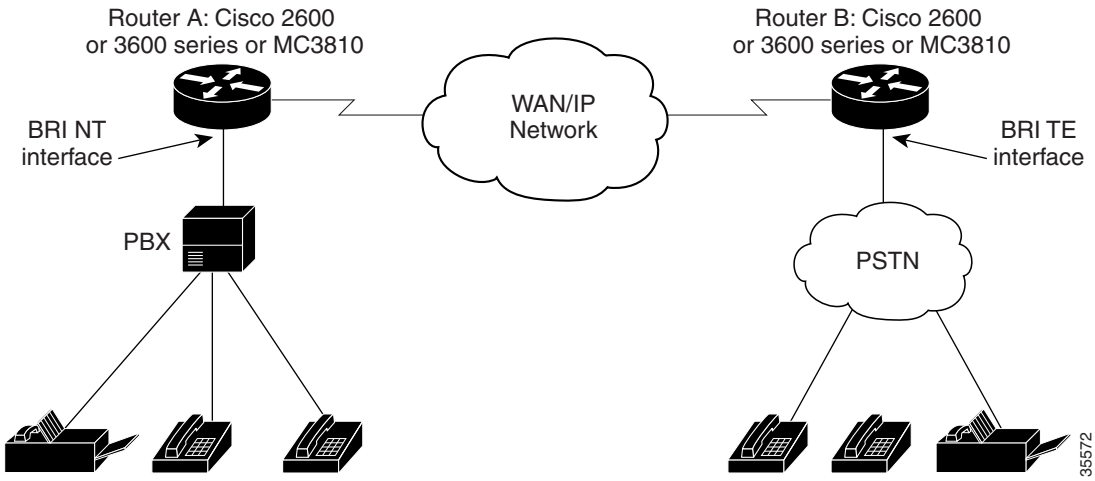
ISDN PRI provides 23 B channels plus a D channel (in North America and Japan) or 30 B channels plus a D channel (in the rest of the world). Similar to the ISDN BRI D channel, the ISDN PRI D channel carries signaling traffic. ISDN PRI is often referred to as “23 B + D” (in North America and Japan) or “30 B + D” (in the rest of the world). The D channel notifies the central office switch to send the incoming call to particular time slots on the Cisco access server or router. Each one of the B channels carries data or voice. The D channel carries signaling for the B channels. The D channel identifies if the call is a circuit-switched digital call or an analog modem call. Analog modem calls are decoded and then sent to the onboard modems. Circuit-switched digital calls are relayed directly to the ISDN processor in the router.

The ISDN BRI NT/TE voice interface card (VIC-2BRI-NT/TE) for the Cisco 2600 and Cisco 3600 series routers and the ISDN BRI voice module (BVM4-NT/TE) for the Cisco MC3810 multiservice concentrator enable Cisco IOS software to replicate the PSTN interface to a PBX that is compatible with European Telecommunications Standards Institute (ETSI) NET3 and QSIG switch types.

Prior to the release of these voice network modules and interface cards, customers with PBXs that implement only the BRI TE interface had to make substantial hardware and software changes on the PBX to implement the NT interface. The implementation of an NT interface on the router allows the customer to connect ISDN PBXs and key systems to a multiservice network with a minimum of configuration changes on the PBX.

The typical application (see Figure 125) allows enterprise customers with a large installed base of legacy telephony equipment to bypass the PSTN.

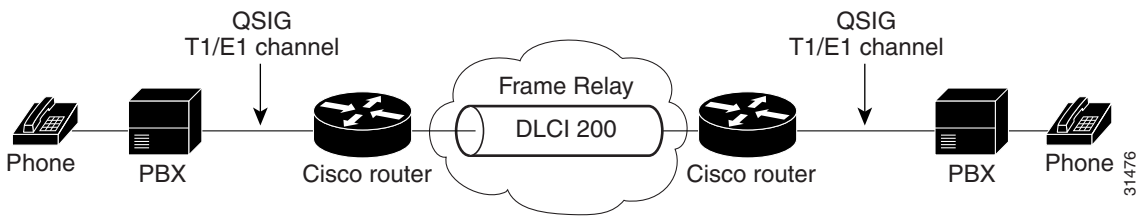
Figure 125 Typical Application Using ISDN BRI NT/TE VICs or ISDN BVMs



QSIG Protocol Support

Integration of QSIG protocol support with Cisco voice switching services allows Cisco devices to connect PBXs, key systems (KTs), and central office switches (COs) that communicate by using the QSIG protocol. The QSIG protocol is becoming the standard for PBX interoperability in Europe and North America. QSIG is a variant of ISDN D-channel voice signaling that is based on the ISDN Q.921 and Q.931 standards. With QSIG, Cisco networks emulate the functionality of the PSTN, and QSIG signaling messages allow the dynamic establishment of voice connections across a Cisco WAN to a peer router, which can then transport the signaling and voice packets to a second PBX, as shown in Figure 126.

Figure 126 QSIG Signaling



The Cisco voice packet network appears to the traditional QSIG PBXs as a distributed transit PBX that can establish calls to any PBX, non-QSIG PBX, or other telephony endpoint served by a Cisco gateway, including non-QSIG endpoints.

When QSIG messages originate and terminate on QSIG endpoints, the QSIG messages are passed transparently across the network; the PBXs are responsible for processing and provisioning the supplementary services. When QSIG and non-QSIG endpoints are linked via a Cisco packet voice gateway, only basic calls are supported. In addition, all switched voice connections must be established and torn down in response to QSIG control messages.

QSIG voice signaling provides the following benefits:

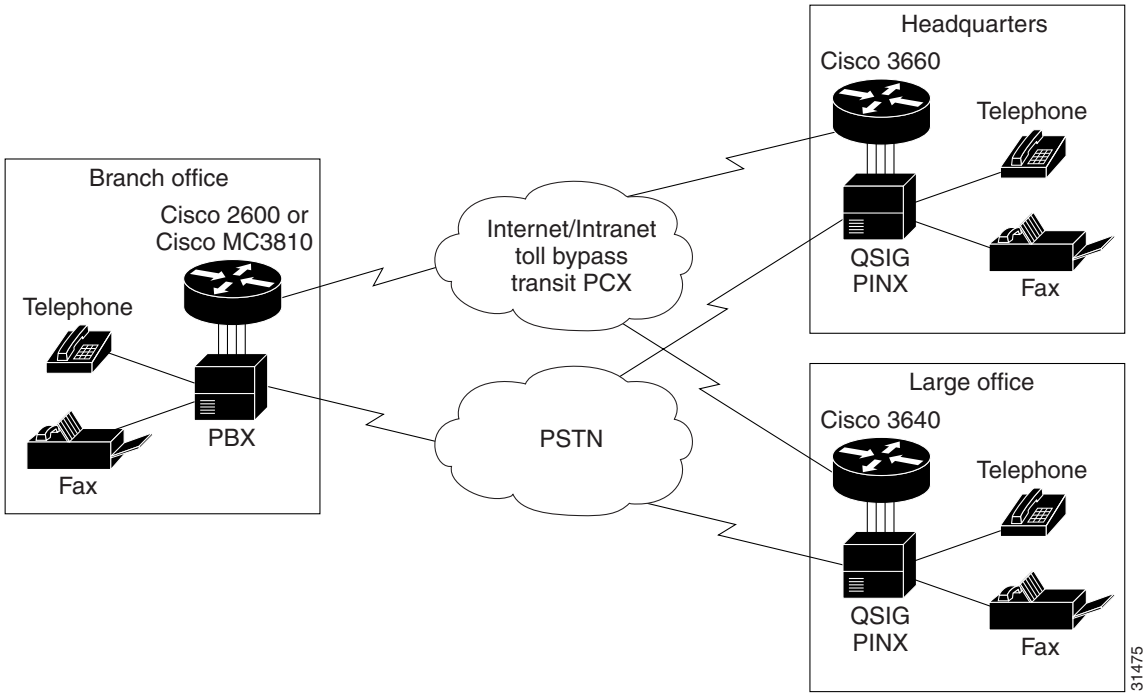
- It provides efficient and cost-effective services on permanent (virtual) circuits or leased lines.
- It allows enterprise networks that include PBX networks to replace leased voice lines with a Cisco WAN.
- It eliminates the need to route connections through multiple tandem PBX hops to reach the desired destination, thereby saving bandwidth, PBX hardware, and switching power.
- It improves voice quality through the single-hop routing provided by voice switching while allowing voice to be compressed more aggressively, resulting in additional savings.
- It supports PBX feature transparency across a WAN, permitting PBX networks to provide advanced features such as calling name and number display, camp-on/callback, network call forwarding, centralized attendant, and centralized message waiting. Usually these capabilities are available on only a single site where users are attached to the same PBX.

QSIG support includes the following capabilities:

- It enables digit forwarding on POTS dial peers.
- On Cisco 2600 series routers, it enables QSIG-switched calls over Voice over Frame Relay (VoFR) and Voice over IP (VoIP) for T1/E1 and BRI voice interface cards.
- On Cisco 3600 series routers, it enables QSIG-switched calls over VoFR, VoIP, and Voice over ATM (VoATM) for T1/E1 and BRI voice interface cards.
- On Cisco 7200 series routers, it enables QSIG-switched calls over VoFR and VoIP on T1/E1 voice interface cards.
- On Cisco MC3810 multiservice concentrators, it enables T1 or E1 PRI and BRI QSIG-switched calls over VoFR, VoIP, and VoATM for Cisco MC3810 digital voice modules (DVMs) and BRI voice module (BVM). QSIG support on the Cisco MC3810 multiservice concentrator was introduced in Cisco IOS Release 12.0(2)T.

Figure 127 shows an example of how QSIG support can enable a toll-bypass application.

Figure 127 QSIG Toll-Bypass Application



QSIG Protocol Stack

QSIG is a variant of ISDN D-channel signaling. The protocol was originally specified by European Computer Manufacturers Association (ECMA), and then was adopted by European Telecommunications Standards Institute (ETSI) and the International Organization for Standardization (ISO). Table 49 identifies the ECMA standards and the OSI layer of the QSIG protocol stack to which they relate.

Table 49 QSIG Protocol Stack

Layer	Standards	Description
Layers 4 to 7	Application mechanisms	End-to-end protocols; network transparent
Layer 3	Multiple ECMA standards	Standards for supplementary services and advance network features
	ECMA-165	QSIG generic functional procedures
	ECMA-142/143	QSIG basic call
Layer 2	ECMA-141	Interface-dependent protocols
Layer 1	I.430 / I.431	PRI and BRI

Switch-Type Configuration Options

To support QSIG at either the global configuration level or the interface configuration level, use the **isdn switch-type** command. For example, if you have a QSIG connection on one line and on the BRI or PRI port, you can configure the ISDN switch type in one of the following combinations:

- Set the global **isdn switch-type** command to support QSIG by entering either the **isdn switch-type basic-qsig** command (BRI) or **isdn switch-type primary-qsig** command (PRI); and set the interface **isdn switch-type** command for the interfaces to a regular central office switch type such as those shown in Table 50.
- Set the global **isdn switch-type** command to support the CO switch type (see Table 50), and set the interface **isdn switch-type** command for the interface to support QSIG.
- Configure the global **isdn switch-type** command to another setting (see Table 50); then set the interface **isdn switch-type** command for **interface bri** to a BRI setting; set the interface **isdn switch-type** command for the serial interface to support QSIG.

Table 50 ISDN CO Switch Types

Country	ISDN Switch Type	Description
Australia	basic-ts013	Australian TS013 switches
Europe	basic-1tr6	German 1TR6 ISDN switches
	basic-nwnet3	Norwegian NET3 ISDN switches (phase 1)
	basic-net3	NET3 ISDN switches (United Kingdom and others)
	vn2	French VN2 ISDN switches
	vn3	French VN3 ISDN switches
Japan	ntt	Japanese NTT ISDN switches
New Zealand	basic-nznet3	New Zealand NET3 switches
North America	basic-5ess	Lucent Technologies basic rate switches
	basic-dms100	NT DMS-100 basic rate switches
	basic-ni1	National ISDN-1 switches

Q.931 Support

Cisco platforms that support Q.931 offer both user- and network-side switch types for ISDN call processing, providing the following benefits:

- User-side PRI enables the Cisco platform to provide a standard ISDN PRI user-side interface to the PSTN.
- Network-side PRI enables the Cisco platform to provide a standard ISDN PRI network-side interface via digital T1/E1 packet voice trunk network modules on Cisco 2600 series and Cisco 3600 series routers.

ISDN Voice Interface Limitations

- Basic-net3 and basic-qsig are the only ISDN switch types currently supported for an NT interface.
- When the ISDN BRI port on the router is configured as an NT port, a “rolled” cable (one with the transmit and receive leads swapped) is needed to connect to a TE interface.
- Layer 1 can be configured only as point-to-point (that is, with one TE connected to each NT). Automatic TEI support will issue only one TEI.

QSIG Support Limitations

The Cisco 2600 series routers do not support VoATM.

The following restrictions apply to the Cisco MC3810 multiservice concentrator:

- QSIG data calls are not supported. All calls with bearer capability indicating a nonvoice type (such as for video telephony) are rejected.
- A Cisco MC3810 multiservice concentrator supports only one T1/E1 interface with direct connectivity to a private integrated services network exchange (PINX).
- The Cisco MC3810 multiservice concentrator supports a maximum of 24 B channels.
- When QSIG is configured, serial port 1 cannot support speeds higher than 192 kbps. This restriction assumes that the MFT is installed in slot 3 on the Cisco MC3810 multiservice concentrator. If the MFT is not installed, then serial port 1 does not operate.

The following restrictions apply to the Cisco 7200 series routers:

- VoATM is not supported.
- BRI is not supported.

ISDN Voice Interface Prerequisite Tasks

Before you can configure a voice interface for ISDN, you must do the following:

- Obtain PRI or BRI service and T1 or E1 service from your service provider, as required. Any BRI lines must be provisioned at the switch to support voice calls.
- Establish a working IP, Frame Relay, or ATM network. At least one network module or WAN interface card must be installed in the router to provide the connection to the LAN or WAN. For more information on installing network modules and interface cards, see the list of documents at the beginning of this chapter.
  - For more information about configuring IP, see the chapter "Voice over IP Overview."
  - For more information about configuring Frame Relay, see the chapter "Configuring Voice over Frame Relay."
  - For more information about configuring ATM, see the chapter "Configuring Voice over ATM."
- Complete your company's dial plan.
- Establish a working telephony network based on your company's dial plan and configure the network for real-time voice traffic. This chapter describes only a portion of the process; for further information, see the chapter "Cisco Voice Telephony."
- Cisco 2600 and Cisco 3600 Series Routers—Install digital T1 or E1 packet voice trunk network modules, BRI voice interface cards, and other voice interface cards as required on your network.
- Cisco 7200 Series Routers—Install a single-port 30-channel T1/E1 high-density voice port adapter.
- Cisco MC3810 Multiservice Concentrators—Install the required digital voice modules (DVMs), BRI voice module (BVM), and multiflex trunk modules.
- All Platforms (As Required):
  - Configure voice card and controller settings.
  - Configure serial and LAN interfaces.
  - Configure voice ports.
  - Configure voice dial peers.

ISDN Voice Interface Configuration Task List

To configure your router for ISDN voice interface support, perform the tasks described in the following sections:

- Configuring ISDN BRI Interfaces, page 629 (required for BRI)
- Configuring ISDN PRI Interfaces, page 636 (required for PRI)

To configure your router for QSIG support, perform the tasks described in the following sections:

- Configuring Global QSIG Support for BRI or PRI, page 638 (required)
- Configuring Controllers for QSIG over PRI, page 639 (required for PRI)
- Configuring BRI Interfaces for QSIG, page 640 (required for BRI)
- Configuring PRI Interfaces for QSIG, page 642 (required for PRI)

To configure your router for Q.931 support, perform the tasks described in the following section:

- Configuring ISDN PRI Q.931 Support, page 648 (required)

Configuring ISDN BRI Interfaces

The steps in this section include commands for configuring an NT interface and a TE interface. To configure an ISDN BRI interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type switch-type`	Configures the telephone company ISDN switch type. For a list of switch types, see Table 51. **Note** The only switch types currently supported for an NT interface are basic-net3 and basic-qsig.

Step 2	**Cisco MC3810 Multiservice Concentrators**
	`Router(config)# interface bri number`
	Other Supported Routers
	`Router(config)# interface bri slot|port`
		Enters interface configuration mode to configure parameters for the specified interface.
		The arguments are as follows:
		• *number*—Specifies the voice module (from 1 to 4).
		• *slot*—Specifies the location of the voice network module in the router (from 1 to 6).
		• *port*—Specifies the location of the BRI voice interface card (VIC) in the voice network module. Valid values are 1 or 2.

Step 3	`Router(config-if)# no ip address`	Specifies that there is no IP address for this interface. For information about IP addressing, refer to the *Cisco IOS IP Configuration Guide*.
Step 4	`Router(config-if)# isdn overlap-receiving`	(Optional) Activates overlap signaling to send to the destination PBX. In this mode, the interface waits for possible additional call-control information.
Step 5	`Router(config-if)# isdn twait-disable`	(Optional) Delays a National ISDN BRI switch for a random length of time before activating the Layer 2 interface when the switch starts up. Use this command when the ISDN switch type is basic-ni1.
Step 6	`Router(config-if)# isdn spid1 spid-number [ldn]`	(Optional; TE only) Specifies a service profile identifier (SPID) and optional local directory number for the B1 channel. Currently, only the DMS-100 and NI-1 switch types require SPIDs. Although some switch types might support a SPID, Cisco recommends that you set up ISDN service without SPIDs.
Step 7	`Router(config-if)# isdn spid2 spid-number [ldn]`	(Optional; TE only) Specifies a SPID and optional local directory number for the B2 channel.
Step 8	`Router(config-if)# isdn incoming-voice voice`	Configures the port for incoming voice calls.
Step 9	`Router(config-if)# shutdown`	Turns off the port (prior to setting the port emulation).

	Command	Purpose
Step 10	`Router(config-if)# isdn layer1-emulate {user | network}`	Configures the Layer 1 port mode emulation and clock settings. The keywords are as follows: • **user**—Configures the port as TE and sets it to function as a clock slave. This is the default. • **network**—Configures the port as NT and sets it to function as a clock master.
Step 11	`Router(config-if)# no shutdown`	Turns on the port.
Step 12	`Router(config-if)# network-clock-priority {low | high}`	(Optional; TE only) Configures the priority of the network clock for this BRI voice port. If this port is configured as TE and you want it to be the first-priority BRI voice port for recovering the clock signal from the network NT device, enter **high**. If this BRI voice port is configured as TE and you want it to be a low-priority BRI voice port for recovering the clock signal from the network NT device, enter **low**. The default for the BRI voice module (BVM) is **low.** The default for the BRI VIC is **high**. Do not use this command if this port is configured as NT in Step 10 with the command **isdn layer1-emulate network**.
Step 13	**Cisco MC3810 Multiservice Concentrators Only** `Router(config-if)# [no] line-power`	Controls the power supplied from an NT-configured port to a TE device. The **line-power** command turns the port power on; the **no line-power** command turns it off. The default is **no line-power**.
Step 14	`Router(config-if)# isdn protocol-emulate {user | network}`	Configures the Layer 2 and Layer 3 port protocol emulation. The keywords are as follows: • **user**—Configures the port as TE; the PBX is the master. This is the default. • **network**—Configures the port as NT; the PBX is the slave.
Step 15	`Router(config-if)# isdn sending-complete`	(Optional) Configures the voice port to include the "Sending Complete" information element in the outgoing call setup message. This command is used in some geographic locations, such as Hong Kong and Taiwan, where the "Sending Complete" information element is required in the outgoing call setup message.

	Command	Purpose
Step 16	`Router(config-if)# isdn static-tei tei-number`	(Optional) Configures a static ISDN Layer 2 terminal endpoint identifier (TEI). The value of *tei-number* can be from 0 to 64.
Step 17	`Router(config-if)# isdn point-to-point-setup`	(Optional) Configures the ISDN port to send SETUP messages on the static TEI. **Note** A static TEI must be configured in order for this command to be effective.
Step 18	`Router(config-if)# end`	Exits interface configuration mode.

Step 19	**Cisco MC3810 Multiservice Concentrators**
	`Router(config)# clear interface bri number`
	Other Supported Routers
	`Router# clear interface slot|port`
		(Optional) Resets the specified interface. The interface needs to be reset if the static TEI number has been configured in Step 16.
		The arguments are as follows:
		• *number*—Specifies the voice module (from 1 to 4).
		• *slot*—Specifies the location of the voice network module in the router (from 1 to 6).
		• *port*—Specifies the location of the BRI VIC in the voice network module. Valid values are 1 or 2.

When you have finished configuring one interface, you must repeat the appropriate steps above for the other interfaces.

Note To complete voice configuration, you must set up your voice ports and dial peers. To do this, see the chapter "Configuring Voice Ports."

Table 51 lists the ISDN switch types.

Table 51 ISDN Switch Types

ISDN Switch Type	Description
basic-qsig	PINX (PBX) switches with QSIG signaling in compliance with Q.931
basic-ts013	Australian TS013 switches
basic-1tr6	German 1TR6 ISDN switches
basic-nwnet3	Norwegian NET3 ISDN switches (phase 1)
basic-net3	NET3 (TBR3) ISDN, Norway NET3, and New Zealand NET3 switches. (This switch type covers the Euro-ISDN E-DSS1 signaling system and is ETSI-compliant.)
vn2	French VN2 ISDN switches
vn3	French VN3 ISDN switches
ntt	Japanese NTT ISDN switches
basic-nznet3	New Zealand NET3 switches

Table 51 ISDN Switch Types (continued)

ISDN Switch Type	Description
basic-5ess	Lucent Technologies basic rate switches
basic-dms100	NT DMS-100 basic rate switches
basic-ni1	National ISDN-1 switches

Verifying ISDN BRI Interface Configuration

To verify the ISDN BRI interface configuration, perform the following steps:

Step 1 Enter the **show running-config** command in EXEC mode to show the current configuration running on the router.

Note The examples show some of the command output that is relevant to BRI configuration tasks. The first example is from a Cisco 2600 series router.

```
Router# show running-config

Building configuration...
Current configuration:
!
version 12.2
!
no service udp-small-servers
service tcp-small-servers
!
hostname Router
!
username xxxx password x 11x5xx07
no ip domain-lookup
ip host Labhost 172.22.66.11
ip host Labhost2 172.22.66.12
ip name-server 172.22.66.21
!
.
.
.
interface BRI1/0
no ip address
no ip directed-broadcast
isdn switch-type basic-net3
isdn overlap-receiving
isdn T306 30000
isdn skipsend-idverify
isdn incoming-voice voice
!
interface BRI1/1
no ip address
no ip directed-broadcast
isdn switch-type basic-net3
isdn overlap-receiving
isdn T306 30000
isdn skipsend-idverify
isdn incoming-voice voice
!
```

```
interface BRI2/0
no ip address
isdn switch-type basic-net3
isdn overlap-receiving
isdn protocol-emulate network
isdn layer1-emulate network
isdn T306-30000
isdn sending-complete
isdn skipsend-idverify
isdn incoming-voice voice
!
interface BRI2/1
no ip address
isdn switch-type basic-net3
isdn overlap-receiving
isdn protocol-emulate network
isdn layer1-emulate network
isdn T306-30000
isdn sending-complete
isdn skipsend-idverify
isdn incoming-voice voice
!
.
.
.
```

The following example is from a Cisco MC3810 multiservice concentrator:

```
Router# show running-config

Building configuration...
Current configuration:
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname Router
!
no logging console
!
network-clock base-rate 56k
network-clock-select 2 T1 0
network-clock-select 3 system(SCB)
network-clock-select 1 BVM
ip subnet-zero
!
isdn switch-type basic-net3
isdn voice-call-failure 0
call rsvp-sync
!
voice-card 0
!
controller T1 0
 mode atm
 framing esf
 linecode b8zs
!
interface BRI1
no ip address
isdn switch-type basic-net3
isdn protocol-emulate network
isdn layer1-emulate network
```

```
isdn incoming-voice voice
isdn T306 30000
isdn skipsend-idverify
no cdp enable
!
interface BRI2
no ip address
isdn switch-type basic-net3
isdn protocol-emulate network
isdn layer1-emulate network
isdn incoming-voice voice
isdn T306 30000
isdn skipsend-idverify
no cdp enable
!
interface BRI3
no ip address
shutdown
network-clock-priority low
isdn switch-type basic-net3
isdn T306 30000
no cdp enable
!
interface BRI4
no ip address
shutdown
network-clock-priority low
isdn switch-type basic-net3
isdn T306 30000
no cdp enable
!
.
.
.
```

Step 2 Enter the **show interfaces bri** command to display information about the physical attributes of the ISDN BRI B and D channels. The term *spoofing* means that the interface is presenting itself to the IOS software as operational.

The following is sample output from the **show interfaces bri** command for a BRI voice port on a Cisco 2610 router:

```
router# show interfaces bri 1/0

BRI3/1 is up, line protocol is up (spoofing)
  Hardware is Voice NT or TE BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation VOICE, loopback not set
  Last input 00:00:02, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     26110 packets input, 104781 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 5 interface resets
```

```
     0 output buffer failures, 0 output buffers swapped out
     9 carrier transitions
```

The following is sample output from the **show interfaces bri** command for a BRI voice port on a Cisco MC3810 multiservice concentrator:

```
Router# show interfaces bri 1

BRI1 is up, line protocol is up (spoofing)
  Hardware is BVM
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Last input 19:32:19, output 19:32:27, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/1/16 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     13282 packets input, 53486 bytes, 0 no buffer
     Received 1 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     13292 packets output, 53515 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     33 carrier transitions
```

Monitoring and Maintaining ISDN BRI Interfaces

To monitor ISDN interfaces, use these commands as needed:

Command	Purpose

Cisco MC3810 Multiservice Concentrators
`Router# show controllers bri number`
 	
Other Supported Routers
`Router# show controllers bri slot/port`
	Displays information about the ISDN BRI interface.

Cisco MC3810 Multiservice Concentrators
`Router# show voice-port summary number`
 	
Other Supported Routers
`Router# show voice-port summary slot/port`
	Displays information about the BRI voice ports.

`Router# show isdn {memory | status | timers}`	Displays information about memory, status, and Layer 2 and Layer 3 timers.

Command	Purpose
`Router# debug isdn q921`	Displays data link layer (Layer 2) access procedures that are taking place at the router on the D channel (LAPD) of its ISDN interface. The **no** form of this command disables debugging output.
`Router# debug isdn q931`	Displays information about call setup and teardown of ISDN network connections (Layer 3) between the local router (user side) and the network. The **no** form of this command disables debugging output.

Configuring ISDN PRI Interfaces

With ISDN PRI, signaling in VoIP is handled by ISDN PRI group configuration. After ISDN PRI has been configured, you must enter the **isdn incoming-voice** command on the serial interface (acting as the D channel) to ensure a dial tone.

To configure basic ISDN PRI interface parameters, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type` *switch-type*	Configures the telephone company ISDN switch type. For a list of switch types, see Table 51. **Note** The only switch types currently supported for an NT interface are basic-net3 and basic-qsig.
Step 2	**Cisco AS5800 Access Servers** `Router(config)# controller T1 1/0/0`	Enters controller configuration mode and specifies the T1 0 controller on the T1 card. or
	Cisco AS5800 Access Servers `Router(config)# controller T1 1/0/0:1`	Enters controller configuration mode and specifies the T1 1 controller on the T3 card. or
	Cisco AS5300 Access Servers `Router(config)# controller T1 0`	Enters controller configuration mode and specifies the T1 0 controller.
Step 3	`Router(config-controller)# framing esf`	Defines the framing characteristics.
Step 4	`Router(config-controller)# linecode {ami | b8zs | hdb3}`	Sets the line-encoding method to match that of your telephone company service provider. The keywords are as follows: • **ami**—Specifies alternate mark inversion (AMI) as the line-code type. Valid for T1 or E1 controllers. This is the default for T1 lines. • **b8zs**—Specifies B8ZS as the line-code type. Valid for T1 controller only. • **hdb3**—Specifies high-density bipolar 3 (hdb3) as the line-code type. Valid for E1 controller only. This is the default for E1 lines.

	Command	Purpose
Step 5	`Router(config-controller)# pri-group timeslots range`	Configures the ISDN PRI group. The *range* argument specifies a range of time slots that make up the PRI group. The range is from 1 to 23.
Step 6	`Router(config-controller)# exit`	Exits controller configuration mode and returns to global configuration mode.
Step 7	**Cisco AS5800 Access Servers** `Router(config)# interface Serial1/0/0:23`	Enters interface configuration mode and specifies the first ISDN PRI line on the T1 card. (The ISDN serial interface is the D channel.)
	or	
	Cisco AS5800 Access Servers `Router(config)# interface Serial1/0/0:1:23`	Enters interface configuration mode and specifies the first ISDN PRI line on the T3 card. (The ISDN serial interface is the D channel.)
	or	
	Cisco AS5300 Access Servers `Router(config)# interface Serial0:23`	Enters interface configuration mode and specifies the first ISDN PRI line. (The ISDN serial interface is the D channel.)
Step 8	`Router(config-if)# isdn incoming-voice modem`	Enables incoming ISDN voice calls. The **modem** keyword specifies that incoming voice calls will be handled as modems. **Note** You must use the **modem** keyword to enable voice calls. The **modem** keyword represents bearer capabilities of speech.

Configuring ISDN PRI Voice Ports

Under most circumstances, the default voice port command values are adequate to configure voice ports to transport voice data over your existing IP network. However, because of the inherent complexities of PBX networks, you might need to configure specific voice port values, depending on the specifications of the devices in your telephony network.

To configure specific voice port parameters, see the chapter "Configuring Voice Ports."

For more information on specific voice-port configuration commands and additional voice port commands, refer to the *Cisco IOS Voice, Video, and Fax Command Reference.*

Verifying ISDN PRI Configuration

You can check the validity of your voice port configuration by performing the following tasks:

- To verify that the data configured is correct, use the **show voice port** command.
- If you have not configured your device to support Direct Inward Dialing (DID), dial in to the router and verify that you have a dial tone.
- Enter a dual tone multifrequency (DTMF) digit. If the dial tone stops, you have verified two-way voice connectivity with the router.

ISDN PRI Troubleshooting Tips

If you are having trouble connecting a call and you suspect that the problem is associated with voice port configuration, you can try to resolve the problem by performing the following tasks:

- Ping the associated IP address to confirm connectivity. If you cannot successfully ping your destination, refer to the chapter "Configuring IP" in the *Cisco IOS IP Configuration Guide.*
- Determine if the voice feature card (VFC) has been correctly installed. For more information, refer to *Installing Voice-over-IP Feature Cards in Cisco AS5300 Universal Access Servers*, which came with your voice network module (VNM).
- To learn if the VFC is operational, use the **show vfc slot number** command.
- To view layer status information, use the **show isdn status** command. If you receive a status message stating that Layer 1 is deactivated, make sure the cable connection is not loose or disconnected. (This status message indicates a problem at the physical layer.)
- With T1 lines, determine if your a-law setting is correct. With E1 lines, determine if your u-law setting is correct. To configure both a-law and u-law values, use the **cptone** command. For more information about the **cptone** command, refer to the *Cisco IOS Voice, Video, and Fax Command Reference.*
- If dialing cannot occur, use the **debug isdn q931** command to check the ISDN configuration.

Configuring Global QSIG Support for BRI or PRI

If you need additional guidance regarding switch-type configuration, see the section "Switch-Type Configuration Options." The steps in this section apply to both BRI and PRI, except as noted. To do the global configuration of QSIG signaling on the router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	**BRI Only on Cisco MC3810, 2600, and 3600 Series Routers** `Router(config)# isdn switch-type basic-qsig`	(Optional) Configures the global ISDN switch type.
	PRI Only on Any Supported Router `Router(config)# isdn switch-type primary-qsig`	(Optional) Configures the ISDN switch-type to support QSIG signaling. **Note** You can configure the ISDN switch type by using either this global command or the same command in interface configuration mode, depending on your configuration. If you configure the global **isdn switch-type** command for QSIG support, you do not need to configure the interface **isdn switch-type** command for QSIG. For more information, see "Switch-Type Configuration Options" on page 626. For a list of CO switch types, see Table 50.

	Command	Purpose
Step 2	`Router(config)# dspinterface dspfarm slot/port`	(Cisco 7200 series routers only) Configures the digital signal processor (DSP) farm interface.
Step 3	`Router(config)# card type {t1 | e1} slot`	(Cisco 7200 series routers only) Specifies the card type and slot number. Enter the card type as T1 or E1; specify the slot location by using a value from 0 to 5, depending on your router.

Configuring Controllers for QSIG over PRI

The steps in this section do not apply to BRI. To configure controllers for QSIG signaling over PRI, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} controller_number`	Enters controller configuration mode and specifies the controller. Enter the controller as E1 or T1, specifying 1 for a Cisco MC3810 multiservice concentrator and a *slot/port* location on a Cisco 2600, 3600, or 7200 series router. **Note** On the Cisco MC3810 multiservice concentrator, QSIG is supported only on controller 1.
Step 2	`Router(config-controller)# pri-group timeslots range`	Configures the PRI group for either T1 or E1. The argument is as follows: • *range*—Specifies a range of time slots tht make up the PRI group. For T1, the range is from 1 to 23. For E1, the range is from 1 to 31. You can configure the PRI group to include all available time slots, or you can configure a select group of time slots for the PRI group.

Configuring BRI Interfaces for QSIG

To configure BRI interfaces for QSIG support, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco MC3810 Multiservice Concentrators**
	`Router(config)# interface bri number`
	
	Cisco 2600, and 3600 Series Routers
	`Router(config)# interface bri slot/port`
		Enters interface configuration mode to configure parameters for the specified interface.
		The arguments are as follows:
		• *number*—Specifies the voice module (from 1 to 4).
		• *slot*—Specifies the location of the voice network module in the router (from 1 to 6).
		• *port*—Specifies the location of the BRI VIC in the voice network module. Valid values are 1 and 2.

Step 2	**Cisco MC3810, 2600, and 3600 Series Routers Only** `Router(config-if)# isdn static-tei 0`	This command is required. (In previous releases, it was set automatically when the **isdn switch-type basic-qsig** command was issued.)
Step 3	**Cisco MC3810 Multiservice Concentrators Only** `Router(config-if)# isdn layer1-emulate {user | network}`	Configures the Layer 1 port mode emulation and the clock settings. The keywords are as follows: • **user**—Configures the port as TE and sets it to function as a clock slave. This is the default. The term **user** is equivalent to the QSIG term *slave*. • **network**—Configures the port as NT and sets it to function as a clock master. The term **network** is equivalent to the QSIG term *master.*
Step 4	**Cisco MC3810 Multiservice Concentrators Only** `Router(config-if)# network-clock-priority {low | high}`	(TE only) Configures the priority of the network clock for this BRI voice port. If this port is configured as TE and you want it to be the first-priority BRI voice port for recovering the clock signal from the network NT device, enter **high**. If this BRI voice port is configured as TE and you want it to be a low-priority BRI voice port for recovering the clock signal from the network NT device, enter **low**. Do not use this command if this port is configured as NT in Step 3 with the command **isdn layer1-emulate network**.

	Command	Purpose
Step 5	**Cisco 2600 and 3600 Series Routers Only** `Router(config-if)# isdn incoming-voice voice`	Routes incoming voice calls. This is set for voice-capable BRI interfaces by default, except for Cisco 2600 and 3600 series BRI S/T TE voice interface cards, where, unless this command is used, the **isdn incoming-voice modem** configuration setting is converted to **isdn incoming-voice voice** when it receives an incoming call.
Step 6	`Router(config-if)# isdn sending-complete`	(Optional) Configures the voice port to include the "Sending Complete" information element in the outgoing call setup message. This command is used in some geographic locations, such as Hong Kong and Taiwan, where the "Sending Complete" information element is required in the outgoing call setup message.
Step 7	**Cisco MC3810, 2600, and 3600 Series Routers Only** `Router(config-if)# isdn switch-type basic-qsig`	(Optional) If the service provider switch type for this BRI port is different from the global ISDN switch type, configure the interface ISDN switch type to match the service provider switch type. The interface ISDN switch type overrides the global ISDN switch type on this interface. See the section "Switch-Type Configuration Options."
Step 8	`Router(config-if)# isdn protocol-emulate {user | network}`	Configures the Layer 2 and Layer 3 port protocol emulation. The keywords are as follows: • **user**—Configures the port as TE; the PINX is the master. This is the default. The term **user** is equivalent to the QSIG term *slave*. • **network**—Configures the port as NT; the PINX is the slave. The term **network** is equivalent to the QSIG term *master*. **Note** On the Cisco MC3810 multiservice concentrator, this command replaces the **isdn switch-type** [**primary-qsig-slave** | **primary-qsig-master**] command.

	Command	Purpose
Step 9	`Router(config-if)# isdn overlap-receiving value`	(Optional) Activates overlap signaling to send to the destination PBX. In this mode, the interface waits for possible additional call-control information from the preceding PINX. **Note** You can leave the default mode of *enbloc*, in which all call establishment information is sent in the setup message without need for additional messages from the preceding PINX.
Step 10	`Router(config-if)# isdn network-failure-cause value`	(Optional) Specifies the cause code to pass to the PBX when a call cannot be placed or completed because of internal network failures. Possible values range from 1 to 127.

Configuring PRI Interfaces for QSIG

To configure PRI interfaces for QSIG support, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco MC3810 Multiservice Concentrators**
	`Router(config)# interface serial 1:channelnumber`
	
	Or
	
	Other Supported Routers
	`Router(config)# interface serial slot/port:channelnumber`
		Enters interface configuration mode for the ISDN PRI interface and the specified interface slot location and channel number. Enter the slot location as **1**. For T1, enter the channel number as **23**. For E1, enter **15**.
		
		Enters interface configuration mode for the ISDN PRI interface and the specified interface slot and port location and channel number. Enter a *slot* number from 1 to 6 and a *port* number of 1 or 2. For T1, enter the channel number as **23**. For E1, enter **15**.

Step 2	`Router(config-if)# isdn switch-type primary-qsig`	If you did not configure the global PRI ISDN switch type for QSIG support in global configuration mode, configure the interface ISDN switch type to support QSIG signaling. See the section "Switch-Type Configuration Options." The conditions that apply to this command in global configuration mode also apply to this command in interface configuration mode. **Note** For this interface, this interface configuration command overrides the setting of the **isdn switch-type** command entered in global configuration mode.

	Command	Purpose
Step 3	`Router(config-if)# isdn contiguous-bchan`	(E1 only) Specifies contiguous bearer channel handling so that B channels 1 through 30 map to time slots 1 to 31, skipping time slot 16.
Step 4	`Router(config-if)# isdn protocol-emulate {user | network}`	Configures the Layer 2 and Layer 3 port protocol emulation. The keywords are as follows: • **user**—Configures the port as TE; the PINX is the master. This is the default. The term **user** is equivalent to the QSIG term *slave*. • **network**—Configures the port as NT; the PINX is the slave. The term **network** is equivalent to the QSIG term *master*. **Note** On the Cisco MC3810 multiservice concentrator, this command replaces the **isdn switch-type** [**primary-qsig-slave** | **primary-qsig-master**] command.
Step 5	`Router(config-if)# isdn overlap-receiving value`	(Optional) Activates overlap signaling to send to the destination PBX. In this mode, the interface waits for possible additional call-control information from the preceding PINX. **Note** You can leave the default mode of *enbloc*, in which all call establishment information is sent in the setup message without need for additional messages from the preceding PINX.
Step 6	`Router(config-if)# isdn network-failure-cause value`	(Optional) Specifies the cause code to pass to the PBX when a call cannot be placed or completed because of internal network failures. Possible values range from 1 to 127.

Verifying the QSIG Configuration

To confirm the QSIG configuration, perform the following steps. The **show running-config** command displays PRI time slot group configuration and other details.

Step 1 To see information about switch type, memory, status, and Layer 2 and Layer 3 timers, enter the **show isdn** command.

For more information about this command, refer to the *Cisco IOS Dial Technologies Command Reference*.

The following sample output shows the results of the **show isdn status** command for a BRI voice port on a Cisco 3600 series router:

```
Router# show isdn status

Global ISDN Switchtype = primary-qsig
```

```
ISDN Serial3/1:15 interface
        dsl 0, interface ISDN Switchtype = primary-qsig
         **** Master side configuration ****
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Layer 3 Status:
        29 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 29
        CCB:callid=89BF, sapi=0, ces=0, B-chan=5, calltype=VOICE
.
.
.
CCB:callid=89C8, sapi=0, ces=0, B-chan=14, calltype=VOICE
.
.
.
        CCB:callid=89D9, sapi=0, ces=0, B-chan=1, calltype=VOICE
        CCB:callid=89DA, sapi=0, ces=0, B-chan=2, calltype=VOICE
        CCB:callid=89DB, sapi=0, ces=0, B-chan=3, calltype=VOICE
    The Free Channel Mask: 0x80000018
ISDN Serial3/0:15 interface
        dsl 1, interface ISDN Switchtype = primary-qsig
         **** Master side configuration ****
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
        TEI = 0, Ces = 9, SAPI = 16, State = TEI_ASSIGNED
    Layer 3 Status:
        28 Active Layer 3 Call(s)
    Activated dsl 1 CCBs = 28
        CCB:callid=BDF, sapi=0, ces=0, B-chan=2, calltype=VOICE
        CCB:callid=BE0, sapi=0, ces=0, B-chan=1, calltype=VOICE
        CCB:callid=BE1, sapi=0, ces=0, B-chan=3, calltype=VOICE
.
.
.
CCB:callid=BFA, sapi=0, ces=0, B-chan=31, calltype=VOICE
    The Free Channel Mask: 0xB0000000
    Total Allocated ISDN CCBs = 54

Total Allocated ISDN CCBs = 0
.
.
.
CCB:callid=89C8, sapi=0, ces=0, B-chan=14, calltype=VOICE
.
.
.
        CCB:callid=89D9, sapi=0, ces=0, B-chan=1, calltype=VOICE
        CCB:callid=89DA, sapi=0, ces=0, B-chan=2, calltype=VOICE
        CCB:callid=89DB, sapi=0, ces=0, B-chan=3, calltype=VOICE
    The Free Channel Mask: 0x80000018
ISDN Serial3/0:15 interface
        dsl 1, interface ISDN Switchtype = primary-qsig
         **** Master side configuration ****
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
        TEI = 0, Ces = 9, SAPI = 16, State = TEI_ASSIGNED
```

```
    Layer 3 Status:
        28 Active Layer 3 Call(s)
    Activated dsl 1 CCBs = 28
        CCB:callid=BDF, sapi=0, ces=0, B-chan=2, calltype=VOICE
        CCB:callid=BE0, sapi=0, ces=0, B-chan=1, calltype=VOICE
        CCB:callid=BE1, sapi=0, ces=0, B-chan=3, calltype=VOICE
.
.
.
CCB:callid=BFA, sapi=0, ces=0, B-chan=31, calltype=VOICE
    The Free Channel Mask: 0xB0000000
    Total Allocated ISDN CCBs = 54
```

The following sample output shows the results of the **show isdn status** command for a BRI voice port and a PRI voice port on a Cisco MC3810 multiservice concentrator:

```
Router# show isdn status

Global ISDN Switchtype = basic-qsig
ISDN BRI1 interface
dsl 1, interface ISDN Switchtype = basic-qsig
**** Slave side configuration ****
    Layer 1 Status:
DEACTIVATED
    Layer 2 Status:
TEI = 0, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
    Layer 3 Status:
NLCB:callid=0x0, callref=0x0, state=31, ces=0 event=0x0
0 Active Layer 3 Call(s)
    Activated dsl 1 CCBs = 0
ISDN BRI2 interface
.
.
.
Router# show isdn status

     Global ISDN Switchtype = primary-qsig
     ISDN Serial1:23 interface
             dsl 0, interface ISDN Switchtype = primary-qsig
              **** Slave side configuration ****
         Layer 1 Status:
             DEACTIVATED
         Layer 2 Status:
             TEI = 0, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
         Layer 3 Status:
             0 Active Layer 3 Call(s)
         Activated dsl 0 CCBs = 0
         The Free Channel Mask: 0x7FFFFF
```

The following sample output shows the results of the **show isdn status** command for a PRI voice port on a Cisco 7200 series router:

```
Router# show isdn status

Global ISDN Switchtype = primary-qsig
ISDN Serial1/0:15 interface
        dsl 0, interface ISDN Switchtype = primary-qsig
         **** Slave side configuration ****
    Layer 1 Status:
        DEACTIVATED
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
    Layer 3 Status:
        0 Active Layer 3 Call(s)
```

```
    Activated dsl 0 CCBs = 0
    The Free Channel Mask:  0x7FFF7FFF
ISDN Serial1/1:15 interface
        dsl 1, interface ISDN Switchtype = primary-qsig
         **** Slave side configuration ****
    Layer 1 Status:
        DEACTIVATED
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
    Layer 3 Status:
        0 Active Layer 3 Call(s)
    Activated dsl 1 CCBs = 0
    The Free Channel Mask:  0x7FFF7FFF
    Total Allocated ISDN CCBs = 0
```

Step 2 To display the state and the service status of each ISDN channel, enter the **show isdn service** command in privileged EXEC mode.

The following example shows sample output from the **show isdn service** command when PRI is configured on a T1 controller:

```
Router# show isdn service

PRI Channel Statistics:
ISDN Se0:15, Channel (1-31)
  Activated dsl 8
  State (0=Idle 1=Propose 2=Busy 3=Reserved 4=Restart 5=Maint)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
  Channel (1-31) Service (0=Inservice 1=Maint 2=Outofservice)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Step 3 To display the Call Distributor Application Programming Interface (CDAPI) information, use the **show cdapi** command.

The following sample output shows the results of the **show cdapi** command for a PRI voice port on a Cisco 3660 series router:

```
Router# show cdapi

Registered CDAPI Applications/Stacks
====================================

Application: TSP CDAPI Application Voice
        Application Type(s) : Voice Facility Signaling
        Application Level   : Tunnel
        Application Mode    : Enbloc

Signaling Stack: ISDN
        Interface: Se5/0:15

Signaling Stack: ISDN
        Interface: Se5/1:15

Signaling Stack: ISDN
        Interface: Se6/0:15

Signaling Stack: ISDN
        Interface: Se6/1:15
```

```
CDAPI Message Buffers
=====================

Used Msg Buffers: 0, Free Msg Buffers: 9600
Used Raw Buffers: 0, Free Raw Buffers: 4800
Used Large-Raw Buffers: 0, Free Large-Raw Buffers: 480
```

The following sample output shows the results of the **show cdapi** command for a PRI voice port on a Cisco MC3810 multiservice concentrator:

```
Router# show cdapi

Registered CDAPI Applications/Stacks
====================================

Application: TSP CDAPI Application Voice
        Application Type(s) : Voice Facility Signaling
        Application Level   : Tunnel
        Application Mode    : Enbloc

Signaling Stack: ISDN
        Interface: Se1:15

CDAPI Message Buffers
=====================

Used Msg Buffers: 2, Free Msg Buffers: 1198
Used Raw Buffers: 2, Free Raw Buffers: 598
Used Large-Raw Buffers: 0, Free Large-Raw Buffers: 60
```

QSIG Support Troubleshooting Tips

Table 52 lists **debug** and **show** commands that can help you analyze problems with your QSIG configuration. The documents listed at the beginning of this chapter include information about these commands.

Table 52 QSIG Troubleshooting Commands

Command	Purpose
`Router# show isdn status`	Displays the status of all ISDN interfaces, including active layers, timer information, and switch type settings.
`Router# show controller t1/e1`	Displays information about T1 and E1 controllers.
`Router# show voice port summary`	Displays summary information about voice port configuration.
`Router# show dial-peer voice`	Displays how voice dial peers are configured.
`Router# show cdapi`	Displays the Call Distributor Application Programming Interface (CDAPI) information.
`Router# show call history voice record`	Displays information about calls made to and from the router.
`Router# show rawmsg`	Displays information about any memory leaks.

Table 52 QSIG Troubleshooting Commands (continued)

`Router# debug isdn event`	Displays events occurring on the user side (on the router) of the ISDN interface. The ISDN events that can be displayed are Q.931 events (call setup and teardown of ISDN network connections).
`Router# debug tsp`	Displays information about the telephony service provider (TSP).
`Router# debug cdapi { events | detail }`	Displays information about CDAPI application events, registration, messages, and so on.

Configuring ISDN PRI Q.931 Support

To configure ISDN PRI Q.931 support on a Cisco 2600 or Cisco 3600 series router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type primary-net5`	(Optional; see note.) Selects a service provider switch type that accommodates PRI. **Note** You can configure the ISDN switch type in either global configuration mode or interface configuration mode. If you configure the ISDN switch type here in this step, specify the switch type for all PRI ports. If you configure the ISDN switch type in interface configuration mode, specify the switch type for a single interface. The switch type specified in interface configuration mode for any individual interface overrides the switch type specified in global configuration mode.
Step 2	`Router(config)# controller {T1 | E1} slot/port`	Enters controller configuration mode and configures the T1 or E1 controller at the specified *slot/port* location. Valid values for *slot* and *port* are 0 and 1.
Step 3	`Router(config-controller)# pri-group timeslots range`	Configures the PRI group for either T1 or E1. The arguments are as follows: • *range*—Specifies a range of time slots which make up the PRI group. For T1, the range is from 1 to 23. For E1, the range is from 1 to 31. You can configure the PRI group to include all available time slots, or you can configure a select group of time slots for the PRI group.
Step 4	`Router(config-controller)# exit`	Exits controller configuration mode.

	Command	Purpose
Step 5	`Router(config)# interface serial0/0:n`	Enters interface configuration mode and specifies the D-channel interface. For *n*, the D-channel number, use the following values: • **0:23** on a T1 PRI • **0:15** on an E1 PRI
Step 6	`Router(config-if)# isdn protocol-emulate {user | network}`	Configures the Layer 2 and Layer 3 port protocol emulation. The keywords are as follows: • **user**—Configures the port as a slave. This is the default. • **network**—Configures the port as a master.
Step 7	`Router(config-if)# [no] line-power`	Turns on or turns off the power supplied from an NT-configured port to a TE device. The default is **no line-power**.
Step 8	`Router(config-if)# isdn incoming-voice voice`	Routes incoming ISDN voice calls to the voice module.

ISDN Voice Interface Configuration Examples

This section provides specific configuration examples for ISDN interfaces in the following sections:

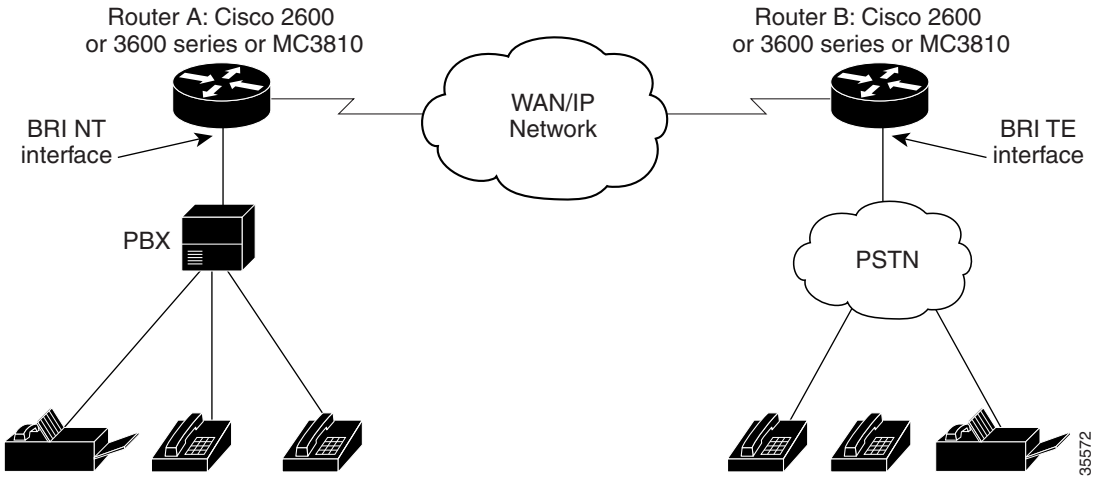
ISDN to PBX and ISDN to PSTN Configuration Examples

This section includes the following configuration examples:

The configuration examples included in this section correspond to the topology shown in Figure 128. The routers each include a BRI VIC and a 2-slot VNM, along with other voice interface cards and modules that are included for completeness. Router A is connected to a PBX through the BRI VIC and is connected to Router B by a serial Ethernet interface. Router B includes a BRI VIC for connection to the PSTN in order to process voice calls from off-premises terminal equipment.

For more information about IP configuration, refer to the *Cisco IOS IP Configuration Guide*. For more information about VoIP, VoFR, and VoATM configuration, see the appropriate configuration information elsewhere in this configuration guide.

Figure 128 Configuration Example Topology



ISDN Connection to a PBX Configuration Example

The following configuration example illustrates the configuration of the BRI interfaces on a Cisco 3640 router (Router A in Figure 128) connected to a PBX:

```
interface BRI1/0
no ip address
isdn switch-type basic-net3
isdn overlap-receiving
isdn protocol-emulate network
isdn layer1-emulate network
isdn T306-30000
isdn sending-complete
isdn skipsend-idverify
isdn incoming-voice voice
!
interface BRI1/1
no ip address
isdn switch-type basic-net3
isdn overlap-receiving
isdn protocol-emulate network
isdn layer1-emulate network
isdn T306-30000
isdn sending-complete
isdn skipsend-idverify
isdn incoming-voice voice
!
ip default-gateway 1.14.0.1
ip classless
ip route 2.0.0.0 255.0.0.0 Ethernet0/1
ip route 2.0.0.0 255.0.0.0 Serial0/1
ip route 172.22.66.33 255.255.255.255 Ethernet0/0
!
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
```

ISDN Connection to the PSTN Configuration Example

The following configuration example illustrates the configuration of the BRI interfaces on a Cisco 2600 series router (Router B in Figure 128) connected to the public ISDN telephone network:

```
interface BRI1/0
no ip address
 no ip directed-broadcast
 isdn switch-type basic-ni1
 isdn twait-disable
 isdn spid1 14085552111 5552111
 isdn spid2 14085552112 5552112
 isdn incoming-voice voice

interface BRI1/1
 no ip address
 no ip directed-broadcast
 isdn switch-type basic-ni1
 isdn twait-disable
 isdn spid1 14085552111 5552111
 isdn spid2 14085552112 5552112
 isdn incoming-voice voice
!
ip classless
ip route 3.0.0.0 255.0.0.0 Ethernet0/1
ip route 3.0.0.0 255.0.0.0 Serial0/1
ip route 172.21.66.0 255.255.255.0 Ethernet0/0
!
!
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
```

QSIG Support Configuration Examples

The following configuration examples shows QSIG configuration on several supported routers:

- QSIG Support on Cisco 3600 Series Routers Example, page 651
- QSIG Support on Cisco 7200 Series Routers Example, page 656
- QSIG Support on Cisco MC3810 Multiservice Concentrators Example, page 661

QSIG Support on Cisco 3600 Series Routers Example

The following configuration example shows how a Cisco 3660 series router can be configured for E1 and PRI with QSIG signaling support using VoIP and VoATM:

```
.
.
.
hostname router3660
!
!
!
!
!
```

```
!
memory-size iomem 20
voice-card 5
!
voice-card 6
!
ip subnet-zero
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller E1 5/0
 pri-group timeslots 1-5,16
!
controller E1 5/1
 pri-group timeslots 1-31
!
controller E1 6/0
 pri-group timeslots 1-31
!
controller E1 6/1
 pri-group timeslots 1-31
!
!
!
interface FastEthernet0/0
 ip address 10.7.72.9 255.255.255.0
 speed auto
 half-duplex
!
interface FastEthernet0/1
 ip address 10.100.100.7 255.255.255.0
 no keepalive
 duplex auto
 speed auto
 hold-queue 1000 in
!
interface Serial2/0
 no ip address
 shutdown
!
interface Serial2/1
 no ip address
 shutdown
!
interface Serial2/2
 no ip address
 shutdown
!
interface Serial2/3
 no ip address
 shutdown
!
interface ATM3/0
 no ip address
 atm clock INTERNAL
 no atm ilmi-keepalive
 pvc 10/40
  vbr-rt 155000 50000 64000
  encapsulation aal5mux voice
 !
```

```
interface Serial5/0:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn overlap-receiving
 isdn incoming-voice voice
 no cdp enable
!
interface Serial5/1:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn incoming-voice voice
 fair-queue 64 256 0
 no cdp enable
!
interface Serial6/0:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn incoming-voice voice
 fair-queue 64 256 0
 no cdp enable
!
interface Serial6/1:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn incoming-voice voice
 fair-queue 64 256 0
 no cdp enable
!
ip classless
ip route 192.168.17.125 255.255.255.255 FastEthernet0/0
no ip http server
!
!
map-class frame-relay frs0
 frame-relay voice bandwidth 1260000
 frame-relay fragment 200
 no frame-relay adaptive-shaping
 frame-relay cir 1260000
 frame-relay fair-queue
!
voice-port 1/0/0
 modem passthrough system
 timing hookflash-in 0
!
voice-port 1/0/1
 modem passthrough system
 timing hookflash-in 0
!
voice-port 5/0:15
 compand-type a-law
!
voice-port 5/1:15
 compand-type a-law
 cptone DE
!
voice-port 6/0:15
```

```
 compand-type a-law
 cptone DE
!
voice-port 6/1:15
 no echo-cancel enable
 compand-type a-law
 cptone DE
!
dial-peer voice 1 pots
 shutdown
 destination-pattern 21...
 modem passthrough system
 direct-inward-dial
!
dial-peer voice 51 voip
 shutdown
 destination-pattern 6504007
 modem passthrough system
 session target ipv4:100.100.100.3
!
dial-peer voice 2 pots
 shutdown
 destination-pattern 21...
 modem passthrough system
 direct-inward-dial
 port 5/1:15
!
dial-peer voice 3 voip
 shutdown
 destination-pattern 22...
 modem passthrough system
 session target ipv4:100.100.100.6
!
dial-peer voice 5 pots
 shutdown
 destination-pattern 22...
 modem passthrough system
 direct-inward-dial
 prefix 4006
!
dial-peer voice 13 pots
 shutdown
 destination-pattern 21...
 modem passthrough system
 direct-inward-dial
 port 6/0:15
!
dial-peer voice 6 pots
 destination-pattern 21...
 modem passthrough system
 direct-inward-dial
 port 6/1:15
!
dial-peer voice 44 voatm
 destination-pattern 22...
 modem passthrough system
 session target ATM3/0 pvc 10/40
!
dial-peer voice 20 pots
 incoming called-number 4...
 destination-pattern 4007
 modem passthrough system
 direct-inward-dial
 port 5/0:15
```

```
 prefix 4007
!
dial-peer voice 21 pots
 destination-pattern 4006
 modem passthrough system
 direct-inward-dial
 port 5/0:15
 prefix 4006
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

QSIG Support on Cisco 7200 Series Routers Example

The following configuration examples show how QSIG protocol support is configured with VoFR on Router A, where calls are originated, and Router B, where calls terminate:

Router A: Originating Configuration

```
.
.
.
hostname 7200_RouterA
!
card type e1 3
card type e1 4
!
!
dspint DSPfarm3/0
!
dspint DSPfarm4/0
!
ip subnet-zero
no ip domain-lookup
ip host routerC 192.168.17.125
ip host routerD 10.1.1.2
!
multilink virtual-template 1
frame-relay switching
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
voice class codec 1
 codec preference 1 g711ulaw
 codec preference 3 g729br8
!
controller E1 3/0
 pri-group timeslots 1-31
 description qsig connected to PCG 1
!
controller E1 3/1
 pri-group timeslots 1-31
 description cas connected to PCG 2
!
controller E1 4/0
 pri-group timeslots 1-31
 description qsig group connected PCG slot3
!
controller E1 4/1
 pri-group timeslots 1-31
 description qsig group connected PCG slot4
!
!
!
!
!
```

Router B: Terminating Configuration

```
.
.
.
hostname 7200_RouterB
!
card type e1 3
card type e1 4
!
!
dspint DSPfarm3/0
!
dspint DSPfarm4/0
!
ip subnet-zero
ip cef
no ip domain-lookup
ip host routerC 192.168.17.125
!
multilink virtual-template 1
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
!
!
controller E1 3/0
 pri-group timeslots 1-31
 description qsig connected to PCG 5
!
controller E1 3/1
 pri-group timeslots 1-31
 description cas connected to PCG 6
!
controller E1 4/0
 pri-group timeslots 1-31
 description cas connected to PCG slot7
!
controller E1 4/1
 pri-group timeslots 1-31
 description cas connected to PCG slot8
!
interface Loopback0
 no ip address
 no ip directed-broadcast
!
```

Router A: Originating Configuration	Router B: Terminating Configuration

interface FastEthernet0/0
 no ip address
 no ip directed-broadcast
 shutdown
 half-duplex
!
!
!
!
!
interface Serial1/0
 bandwidth 512
 ip address 10.1.1.104 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 no keepalive
 shutdown
 no fair-queue
 clockrate 2015232
 ppp multilink
!
interface Serial1/1
 description vofr connection to
7200_RouterB_s1/1
 ip address 10.0.0.2 255.0.0.0
 ip broadcast-address 10.0.0.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip route-cache
 no ip mroute-cache
 no keepalive
 frame-relay traffic-shaping
 frame-relay map ip 10.0.0.1 100 broadcast
 frame-relay interface-dlci 100
  class vofr_class
  vofr data 4 call-control 5
!
interface Serial1/2
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
!
interface Serial1/3
 no ip address
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 shutdown
 clockrate 2015232
!
	interface FastEthernet0/0
	 description VOIP_10.0.0.1_maxstress to
	7200_RouterAgate
	 ip address 10.0.0.1 255.0.0.0
	 no ip directed-broadcast
	 no ip mroute-cache
	 shutdown
	 media-type MII
	 full-duplex
	!
	interface Serial1/0
	 no ip address
	 no ip directed-broadcast
	 no ip mroute-cache
	 shutdown
	!
	!
	!
	!
	!
	!
	!
	!
	!
	interface Serial1/1
	 description vofr connection to
	7200_RouterA
	 ip address 10.0.0.1 255.0.0.0
	 ip broadcast-address 10.0.0.0
	 no ip directed-broadcast
	 encapsulation frame-relay
	 no keepalive
	 clockrate 8060928
	 frame-relay traffic-shaping
	 frame-relay map ip 10.0.0.2 100 broadcast
	 frame-relay interface-dlci 100
	  class vofr_class
	  vofr data 4 call-control 5
	!
	!
	interface Serial1/2
	 no ip address
	 no ip directed-broadcast
	 shutdown
	 clockrate 2015232
	!
	!
	interface Serial1/3
	 no ip address
	 no ip directed-broadcast
	 shutdown
	!
	!
	!
	!

Router A: Originating Configuration	Router B: Terminating Configuration
interface Ethernet2/0 ip address 10.1.50.77 255.255.0.0 ip broadcast-address 10.1.0.0 no ip directed-broadcast no ip route-cache no ip mroute-cache ! interface Ethernet2/1 ip address 10.0.0.2 255.255.0.0 ip broadcast-address 10.0.0.0 no ip directed-broadcast no ip route-cache no ip mroute-cache shutdown ! interface Ethernet2/2 no ip address no ip directed-broadcast no ip route-cache no ip mroute-cache shutdown ! interface Ethernet2/3 no ip address no ip directed-broadcast no ip route-cache no ip mroute-cache shutdown ! interface Serial3/0:15 no ip address no ip directed-broadcast no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! ! ! interface Serial3/1:15 no ip address no ip directed-broadcast no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! ! !	interface Ethernet2/0 ip address 10.5.192.123 255.255.0.0 ip helper-address 192.168.17.125 no ip directed-broadcast no ip mroute-cache ! ! interface Ethernet2/1 ip address 10.0.0.1 255.255.0.0 no ip directed-broadcast no ip mroute-cache shutdown ! ! ! interface Ethernet2/2 no ip address no ip directed-broadcast shutdown ! ! ! interface Ethernet2/3 no ip address no ip directed-broadcast shutdown ! ! ! interface Serial3/0:15 no ip address no ip directed-broadcast no ip route-cache cef ip mroute-cache no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! interface Serial3/1:15 no ip address no ip directed-broadcast no ip route-cache cef ip mroute-cache no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable !

Router A: Originating Configuration	Router B: Terminating Configuration
interface Serial4/0:15 no ip address no ip directed-broadcast no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! ! ! interface Serial4/1:15 no ip address no ip directed-broadcast no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! ! ! interface ATM5/0 no ip address no ip directed-broadcast no ip route-cache no ip mroute-cache shutdown no atm ilmi-keepalive ! ! ! ! ! interface Virtual-Template1 ip address 10.0.0.2 255.255.255.0 no ip directed-broadcast load-interval 30 fair-queue 64 256 1 ppp multilink ppp multilink fragment-delay 20 ppp multilink interleave ip rtp priority 16384 16383 92 ! router igrp 144 network 10.0.0.0 ! ip default-gateway 10.21.75.10 ip classless no ip http server !	interface Serial4/0:15 no ip address no ip directed-broadcast no ip route-cache cef ip mroute-cache no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! interface Serial4/1:15 no ip address no ip directed-broadcast no ip route-cache cef ip mroute-cache no logging event link-status isdn switch-type primary-qsig isdn overlap-receiving isdn incoming-voice voice isdn bchan-number-order ascending no cdp enable ! interface ATM5/0 no ip address no ip directed-broadcast shutdown no atm ilmi-keepalive ! interface FastEthernet6/0 no ip address no ip directed-broadcast shutdown half-duplex ! interface Virtual-Template1 ip unnumbered Loopback0 no ip directed-broadcast no ip route-cache cef ip mroute-cache ppp multilink ppp multilink fragment-delay 20 ppp multilink interleave ! ! router igrp 144 network 10.0.0.0 ! ! ip classless no ip http server !

Router A: Originating Configuration	Router B: Terminating Configuration
map-class frame-relay vofr_class no frame-relay adaptive-shaping frame-relay cir 4400000 frame-relay bc 1000 frame-relay fair-queue frame-relay voice bandwidth 4000000 frame-relay fragment 256 ! voice-port 3/0:15 compand-type a-law cptone DE ! voice-port 3/1:15 compand-type a-law cptone DE ! voice-port 4/0:15 compand-type a-law cptone DE ! voice-port 4/1:15 compand-type a-law cptone DE ! dial-peer voice 5552222 pots destination-pattern +5552... direct-inward-dial port 3/1:15 prefix 5552 ! dial-peer voice 5551111 vofr destination-pattern +5...... sequence-numbers session target Serial1/1 100 codec g729br8 ! dial-peer voice 5554 pots destination-pattern 5554... direct-inward-dial port 4/1:15 prefix 5554 ! dial-peer voice 5553 pots destination-pattern 5553... direct-inward-dial port 4/0:15 prefix 5553 ! dial-peer voice 5551 pots destination-pattern +5551... direct-inward-dial port 3/0:15 prefix 5551 . . .	map-class frame-relay vofr_class no frame-relay adaptive-shaping frame-relay cir 4400000 frame-relay bc 1000 frame-relay fair-queue frame-relay voice bandwidth 4000000 frame-relay fragment 256 ! voice-port 3/0:15 compand-type a-law ! ! voice-port 3/1:15 compand-type a-law ! ! voice-port 4/0:15 compand-type a-law ! ! voice-port 4/1:15 compand-type a-law ! ! dial-peer voice 5552222 pots destination-pattern +5552... direct-inward-dial port 3/1:15 prefix 6662 ! dial-peer voice 5551111 vofr destination-pattern +5...... sequence-numbers session target Serial1/1 100 codec g729br8 ! dial-peer voice 6661 pots destination-pattern +6661... direct-inward-dial port 3/0:15 prefix 6661 ! dial-peer voice 6663 pots destination-pattern +6663... direct-inward-dial port 4/0:15 prefix 6663 ! dial-peer voice 6664 pots destination-pattern +6664... direct-inward-dial port 4/1:15 prefix 6664 . . .

QSIG Support on Cisco MC3810 Multiservice Concentrators Example

The following configuration example shows how a Cisco MC3810 multiservice concentrator can be configured for E1 and PRI with QSIG signaling support and VoIP and VoFR:

```
.
.
.
hostname Router3810
!
!
!
!
network-clock base-rate 56k
ip subnet-zero
!
isdn switch-type primary-qsig
isdn voice-call-failure 0
!
!
!
!
controller T1 0
 mode atm
 framing esf
 clock source internal
 linecode b8zs
!
controller E1 1
 pri-group timeslots 1-7,16
!
!
!
interface Ethernet0
 ip address 100.100.100.6 255.255.255.0
 no ip directed-broadcast
!
interface Serial0
 bandwidth 2000
 ip address 10.168.14.1 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 no keepalive
 clockrate 2000000
 cdp enable
 frame-relay traffic-shaping
 frame-relay interface-dlci 100
  class frs0
  vofr cisco
!
interface Serial1
 no ip address
 no ip directed-broadcast
 shutdown
!
interface Serial1:15
 no ip address
 no ip directed-broadcast
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn overlap-receiving
```

```
 isdn incoming-voice voice
 fair-queue 64 256 0
 no cdp enable
!
interface ATM0
 no ip address
 no ip directed-broadcast
 ip mroute-cache
 no atm ilmi-keepalive
 pvc 10/42
  encapsulation aal5mux voice
 !
!
interface FR-ATM20
 no ip address
 no ip directed-broadcast
 shutdown
!
no ip http server
ip classless
ip route 223.255.254.0 255.255.255.0 Ethernet0
!
!
map-class frame-relay frs0
 frame-relay voice bandwidth 1260000
 frame-relay fragment 200
 no frame-relay adaptive-shaping
 frame-relay cir 1260000
 frame-relay fair-queue
!
map-class frame-relay frsisco
!
voice-port 1:15
 compand-type a-law
!
dial-peer voice 100 voatm
 shutdown
 destination-pattern 4...
 session target ATM0 pvc 10/42
 codec g729ar8
 no vad
!
dial-peer voice 1 pots
 shutdown
 destination-pattern 3001
!
dial-peer voice 42 vofr
 destination-pattern 4006
 session target Serial0 100
 signal-type ext-signal
!
dial-peer voice 21 pots
 destination-pattern 4007
 direct-inward-dial
 port 1:15
 prefix 4007
!
dial-peer voice 12 voip
 shutdown
 destination-pattern 4006
 session target ipv4:100.100.100.7
.
.
.
```

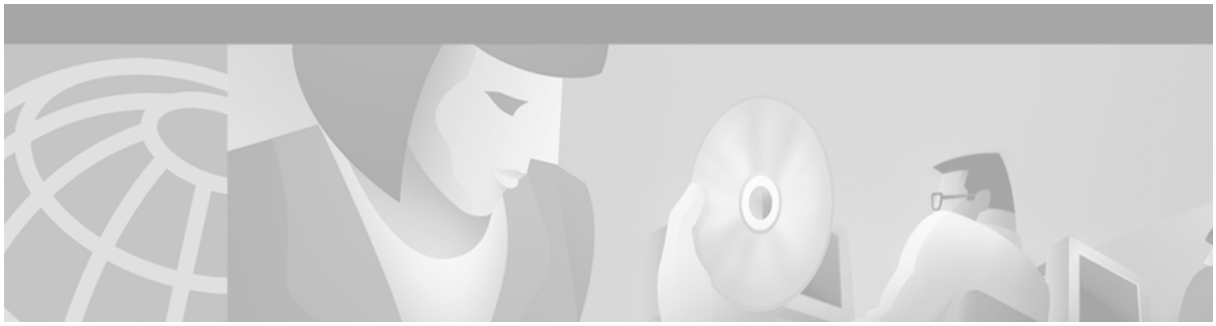
Q.931 Support Configuration Examples

The following configuration example shows how a Cisco 3660 router can be configured for E1 and PRI with network-side support using VoIP:

```
.
.
.
hostname router3660
!
!
memory-size iomem 20
voice-card 5
!
voice-card 6
!
ip subnet-zero
!
isdn switch-type primary-net5
isdn voice-call-failure 0
!
controller E1 3/0
 pri-group timeslots 1-5,16
!
controller E1 3/1
 pri-group timeslots 1-31
!
controller E1 4/0
 pri-group timeslots 1-31
!
controller E1 4/1
 pri-group timeslots 1-31
!
interface FastEthernet0/0
 ip address 10.7.72.9 255.255.255.0
 speed auto
 half-duplex
!
interface FastEthernet0/1
 ip address 10.100.100.7 255.255.255.0
 no keepalive
 duplex auto
 speed auto
 hold-queue 1000 in
!
interface Serial2/0
 no ip address
 shutdown
!
interface Serial2/1
 no ip address
 shutdown
!
interface Serial2/2
 no ip address
 shutdown
!
interface Serial2/3
 no ip address
 shutdown
!
interface Serial5/0:15
 no ip address
```

```
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn overlap-receiving
 isdn incoming-voice voice
 isdn protocol-emulate network
 no cdp enable
!
interface Serial5/1:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn incoming-voice voice
 fair-queue 64 256 0
 no cdp enable
!
interface Serial6/0:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn incoming-voice voice
 fair-queue 64 256 0
 isdn protocol-emulate network
 no cdp enable
!
interface Serial6/1:15
 no ip address
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-qsig
 isdn incoming-voice voice
 fair-queue 64 256 0
 no cdp enable
!
ip classless
ip route 223.255.254.254 255.255.255.255 FastEthernet0/0
no ip http server
!
!
voice-port 1/0/0
timing hookflash-in 0
!
voice-port 1/0/1
timing hookflash-in 0
!
voice-port 5/0:15
 compand-type a-law
!
voice-port 5/1:15
 compand-type a-law
 cptone DE
!
voice-port 6/0:15
 compand-type a-law
 cptone DE
!
voice-port 6/1:15
 no echo-cancel enable
 compand-type a-law
 cptone DE
!
```

```
dial-peer voice 1 pots
 shutdown
 destination-pattern 21...
direct-inward-dial
!
dial-peer voice 51 voip
 shutdown
 destination-pattern 6504007
session target ipv4:100.100.100.3
!
dial-peer voice 2 pots
 shutdown
 destination-pattern 21...
direct-inward-dial
 port 5/1:15
!
dial-peer voice 3 voip
 shutdown
 destination-pattern 22...
session target ipv4:100.100.100.6
!
dial-peer voice 5 pots
 shutdown
 destination-pattern 22...
 modem passthrough system
 direct-inward-dial
 prefix 4006
!
dial-peer voice 13 pots
 shutdown
 destination-pattern 21...
direct-inward-dial
 port 6/0:15
!
dial-peer voice 6 pots
 destination-pattern 21...
direct-inward-dial
 port 6/1:15
!
dial-peer voice 20 pots
 incoming called-number 4...
 destination-pattern 4007
direct-inward-dial
 port 5/0:15
 prefix 4007
!
dial-peer voice 21 pots
 destination-pattern 4006
direct-inward-dial
 port 5/0:15
 prefix 4006
!
!
line con 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```



Configuring PBX Interconnectivity Features

This chapter describes how to configure support for the PBX signaling formats QSIG and Transparent Common Channel Signaling (T-CCS). Configuring support for these signaling protocols on your router enables the router to interoperate with PBXs that are running them.

This chapter includes the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Configuring QSIG PRI Signaling Support

This section describes QSIG PRI voice signaling support for Cisco AS5300 universal access servers and Cisco MC3810 multiservice concentrators.

Benefits of QSIG Voice Signaling

On both the Cisco AS5300 universal access server and the Cisco MC3810 multiservice concentrator, QSIG voice signaling provides the following benefits:

- Enabling Cisco devices to connect with digital PBXs that use the QSIG form of common channel signaling (CCS).
- Access to multiple remote PBXs with a single connection to a Cisco device.
- Transparent support for supplementary PBX services so that proprietary PBX features are not lost when connecting PBXs to Cisco AS5300 and Cisco MC3810 networks.

- QSIG support based on widely used ISDN Q.931 standards. Cisco QSIG implementation follows the following European Telecommunications Standards Institute (ETSI) implementation standards:
  - ECMA-142: *Private Integrated Services Network (PISN) - Circuit Mode 64kbit/s Bearer Services - Service Description, Functional Capabilities and Information Flows (BCSD)*
  - ECMA-143: *PISN - Circuit Mode Bearer Services - Inter-Exchange Signalling Procedures and Protocol (QSIG-BC)* (This specification covers QSIG basic call services.)
  - ECMA-165: PISN - *Generic Functional Protocol for the Support of Supplementary Services - Inter-Exchange Signalling Procedures and Protocol (QSIG-GF)*
- Compatibility with H.323 for IP call setup and transport of QSIG messaging.
- Support for calls that do not require a bearer channel for voice transport.
- Support for bandwidth-on-demand, utilizing network resources only when a connection is desired.

Configuration tasks for QSIG PRI signaling support are described in the following sections:

- Configuring Voice over IP QSIG Network Transparency on the Cisco AS5300, page 668
- Configuring QSIG PRI Signaling Support on the Cisco MC3810, page 673

Although the procedures for configuring QSIG signaling support on the Cisco AS5300 universal access server and on the Cisco MC3810 multiservice concentrator are very similar, implementation differences are described in the respective sections.

Configuring Voice over IP QSIG Network Transparency on the Cisco AS5300

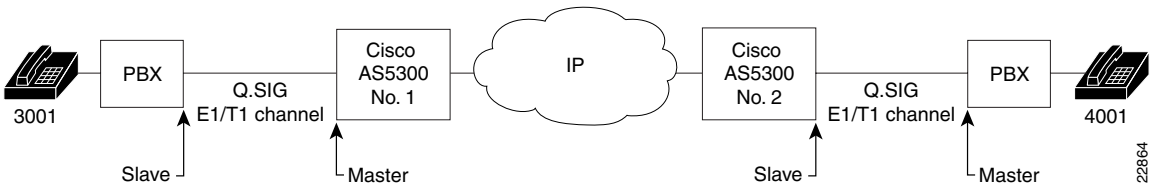
Integration of QSIG support with Voice over IP (VoIP) enables Cisco voice switching services to connect PBXs, key systems, and CO switches that communicate by using the QSIG protocol.

The QSIG protocol is a variant of ISDN D-channel voice signaling. It is based on the ISDN Q.921 and Q.931 standards and is becoming a worldwide standard for PBX interconnection. Using QSIG signaling, Cisco devices can route incoming voice calls from a private integrated services network exchange (PINX) device across a WAN to a peer Cisco device, which can then transport the signaling and voice packets to a second PINX device.

QSIG allows the user to place QSIG calls into and receive QSIG calls from Cisco VoIP networks. The Cisco packet network appears to PBXs as a large, distributed transit PBX that can establish calls to any destination served by a Cisco voice node. The switched voice connections are established and torn down in response to QSIG control messages that come over an ISDN PRI D channel. The QSIG message is passed transparently across the IP network and the message appears to the attached PINX devices as a transit network. The PINX devices are responsible for processing and provisioning the attached services.

Figure 129 shows an example of a QSIG signaling configuration. In this example, the Cisco AS5300 acts either as a master to a slave PBX or as a slave to a master PBX.

Figure 129 Cisco AS5300 QSIG Signaling Configuration



The following restrictions and limitations apply to the Cisco AS5300 QSIG implementation:

- QSIG functionality on the AS5300 requires Cisco IOS Release 12.0(7)T or later and VCWare version 4.04.
- QSIG data calls are not supported. All calls with bearer capability indicating a nonvoice type (such as video telephony) are rejected.
- In order to ensure end-to-end QSIG feature transparency, the incoming POTS dial peer must have DID configured so as to prevent generation of a secondary dial tone.

QSIG Prerequisite Tasks

Perform the following configuration tasks before you configure QSIG for VoIP:

- Configure the ports used on the Cisco AS5300 as voice ports. For information on how to configure ports to be used as voice ports, see the section "Configuring Voice Ports" in the chapter "Configuring Voice over IP."
- Install VCWare version 4.04. For information on how to upgrade or install VCWare, see the section "Managing Cisco AS5300 VFCs" in the chapter "Configuring Voice over IP."
- Configure VoIP. For information on how to configure VoIP, see the chapter "Configuring Voice over IP."

QSIG Configuration Task List

To configure QSIG for Voice over IP, complete the tasks shown in the following sections:

- Configuring VoIP QSIG, page 670 (required)
- Configuring Fusion Call Control Signaling (NEC Fusion) on the Cisco AS5300, page 672 (optional)

Configuring VoIP QSIG

To configure QSIG signaling support on the Cisco AS5300, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type primary-qsig`	(Optional) Globally configures the ISDN switch type to support QSIG signaling. **Note** Depending on your configuration, you can configure the ISDN switch type either by using this command in global configuration mode or by using the same command in interface configuration mode. (See Step 6.) If you configure the ISDN switch type for QSIG support using the **isdn switch-type** command in global configuration mode, you need not use the **isdn switch-type** command in interface configuration mode. If the PBX in your configuration is an NEC PBX, and you are using Fusion Call Control Signaling (FCCS), see the section "Configuring Fusion Call Control Signaling (NEC Fusion) on the Cisco AS5300."
Step 2	`Router(config)# controller {T1 | E1} controller number`	Enters controller configuration mode.
Step 3	`Router(config-controller)# pri-group [timeslot range]`	Configures the PRI group for either T1 or E1 to carry voice traffic. For T1, available time slots are from 1 to 23, and for E1, available time slots are from 1 to 31. You can configure the PRI group to include all available time slots, or you can configure a select group of time slots for the PRI group. For example, if only time slots 1 to 10 are in the PRI group, enter the **pri-group timeslot 1-10** command. If the PRI group includes all channels available for T1 (channels 1 to 23), enter the **pri-group timeslot 1-23** command. If the PRI group includes all channels available for E1 (channels 1 to 31), enter the **pri-group timeslot 1-31** command.
Step 4	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 5	`Router(config)# interface serial 1:channelnumber`	Enters interface configuration mode for the ISDN PRI interface. The *channelnumber* argument specifies the channel number. For T1, enter the channel number as **23**. For E1, enter **15**.

	Command	Purpose
Step 6	`Router(config-if)# isdn switch-type primary-qsig`	(Optional) For the selected interface, configures the ISDN switch type to support QSIG signaling. Use this command if you did not configure the ISDN switch type for QSIG support globally in Step 1. The same conditions that apply to this command in global configuration mode also apply to this command in interface configuration mode. **Note** For the selected interface, this command, entered in interface configuration mode, overrides any setting made with the **isdn switch-type** command entered in global configuration mode.
Step 7	`Router(config-if)# isdn protocol-emulate {user | network}`	Configures the ISDN interface to serve as either the primary QSIG slave or the primary QSIG master. The keywords are as follows: • **user**—Specifies slave. • **network**—Specifies master. If the PINX is the primary QSIG master, configure the Cisco AS5300 to serve as the primary QSIG slave. If the PINX is the primary QSIG slave, configure the Cisco AS5300 to serve as the primary QSIG master.
Step 8	`Router(config-if)# isdn overlap-receiving [T302 value]`	(Optional) Activates overlap signaling to send to the destination PBX. The keyword and argument are as follows: • **T302**—Specifies timer T302. • *value*—Specifies the value of timer T302 in milliseconds. The range is 500 to 20000.
Step 9	`Router(config-if)# isdn incoming-voice modem`	Routes incoming voice calls to the modem and treats them as analog data.
Step 10	`Router(config-if)# isdn network-failure-cause [value]`	(Optional) Specifies the cause code to pass to the PBX when a call cannot be placed or completed because of internal network failures. The *value* argument is a cause code from 1 to 127. All cause codes except Normal Call Clearing (16), User Busy (17), No User Responding (18), and No Answer from User (19) will be changed to the specified cause code.

	Command	Purpose
Step 11	`Router(config-if)# isdn bchan-number-order {ascending | descending}`	(Optional) Configures the ISDN PRI interface to make the outgoing call selection in ascending or descending order. The keywords are as follows: • **ascending**—Makes the outgoing call selection in ascending order. • **descending**—Makes the outgoing call selection in descending order. This is the default. For descending order, the first call from the Cisco AS5300 uses channel 23 (T1) or channel 31 (E1). The second call then uses channel 22 (T1) or channel 30 (E1), and so on, in descending order. For ascending order, if the PRI group starts with 1, the first call uses channel 1, the second call uses channel 2, and so on, in ascending order. If the PRI group starts with a different time slot, the ascending order starts with the lowest time slot.

As shown in the procedure, you have a choice of configuring the **isdn-switch-type** command to support QSIG at either the global configuration level or the interface configuration level. For example, if you have a QSIG connection on one line and on the PRI port, you can configure the ISDN switch type in one of the following combinations:

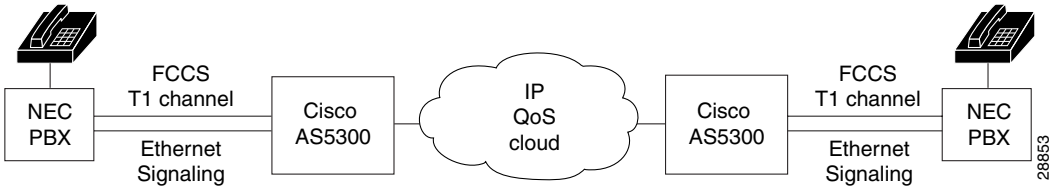
- Set the global **isdn-switch-type** command to support QSIG and set the interface **isdn-switch-type** command for **interface serial 0:23** to a PRI setting such as 5ess.
- Set the global **isdn-switch-type** command to support PRI 5ess and set the interface **isdn-switch-type** command for **interface serial 1:23** to support QSIG.
- Configure the global **isdn-switch-type** command to another setting (such as switch type VN3), set the interface **isdn-switch-type** command for **interface serial 0:23** to a PRI setting, and set the interface **isdn-switch-type** command for **interface serial 1:23** to support QSIG.

Configuring Fusion Call Control Signaling (NEC Fusion) on the Cisco AS5300

If you have an NEC PBX in your network and you are running FCCS, you will need to configure your Cisco AS5300 universal access servers appropriately. FCCS, also known as NEC Fusion, allows individual nodes anywhere within a network to operate as if they were part of a single integrated PBX system. The database storage, share, and access routines of NEC Fusion allow real-time access from any node to any other, enabling individual nodes to learn about the entire network configuration. This capability allows network-wide feature, functional, operational, and administration transparency.

Figure 130 shows an example of a Cisco AS5300 QSIG signaling configuration using an NEC PBX.

Figure 130 QSIG Signaling Configuration with NEC PBX



To configure NEC Fusion signaling support on the Cisco AS5300, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller T1 controller number`	Enters controller configuration mode. **Note** NEC Fusion does not support fractional T1/E1; all 24 channels must be available. If they are not all available, the configuration request will fail.
Step 2	`Router(config-controller)# pri-group nec-fusion {pbx-ip-address/pbx-ip-host-name} pbx-port number`	Configures the controller to communicate with an NEC PBX using NEC Fusion. The *number* argument specifies the PBX port number. The range is 49152 to 65535; the default is 55000. If this value is already in use, the next greater value will be used.

Verifying VoIP QSIG Software on the Cisco AS5300

After you have completed the configuration for the Cisco AS5300, verify that you configured QSIG properly. Enter the **show isdn status** command to view the ISDN layer information. The following output shows that you have correctly designated the global ISDN switch type to be primary-QSIG:

```
Router# show isdn status

Global ISDN Switchtype = primary-qsig
ISDN Serial1:23 interface
        dsl 0, interface ISDN Switchtype = primary-qsig
         **** Slave side configuration ****
    Layer 1 Status:
        DEACTIVATED
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = TEI_ASSIGNED
    Layer 3 Status:
        0 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 0
    The Free Channel Mask: 0x7FFFFF
```

Configuring QSIG PRI Signaling Support on the Cisco MC3810

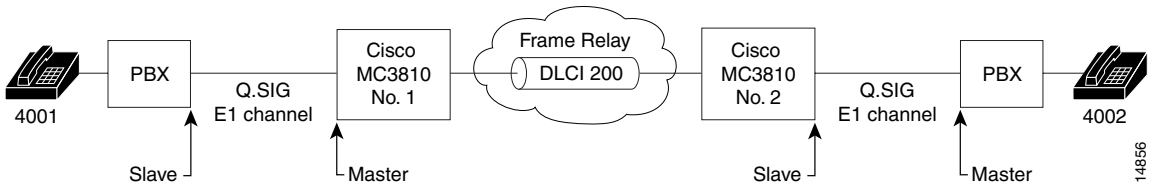
The QSIG protocol provides signaling for PINX devices. It is based on the ISDN Q.931 standard. Using QSIG PRI signaling, the Cisco MC3810 can route incoming voice calls from a PINX device across a WAN to a peer Cisco MC3810, which can then transport the signaling and voice packets to a second PINX device.

The following restrictions and limitations apply to the Cisco MC3810 QSIG PRI implementation:

- QSIG data calls are not supported. All calls with bearer capability indicating a nonvoice type (such as for video telephony) are rejected.
- QSIG is supported only on T1/E1 controller 1. Each Cisco MC3810 supports only one T1/E1 interface with direct connectivity to a PINX device.
- The Cisco MC3810 supports a maximum of 24 bearer channels.
- When QSIG is configured, serial interface 1 cannot support speeds higher than 192 kbps. This restriction assumes that the MFT is installed in slot 3 on the Cisco MC3810. If the MFT is not installed, then serial interface 1 will not operate at all, but QSIG will be supported on other interfaces.

Figure 131 shows an example of a QSIG signaling configuration. In the example, the Cisco MC3810 acts either as a master to a slave PBX or as a slave to a master PBX.

Figure 131 Cisco MC3810 QSIG Signaling Configuration



QSIG Prerequisite Tasks

The following configuration tasks should be completed before you configure QSIG on the Cisco MC3810:

- Configure the ports used on the Cisco MC3810 as voice ports. For information on how to configure ports to be used as voice ports, see the section “Configuring Voice Ports” in the chapter “Configuring Voice over ATM.”
- Configure Voice over Frame Relay or Voice over ATM. For information on how to configure Voice over Frame Relay, see the “Configuring Voice over Frame Relay” chapter. For information on how to configure Voice over ATM, see the “Configuring Voice over ATM” chapter.

To configure QSIG PRI signaling support on the Cisco MC3810, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type primary-qsig`	(Optional) Globally configures the ISDN switch type to support QSIG signaling. **Note** Depending on your configuration, you can configure the ISDN switch type either by using this command in global configuration mode, or by using the same command in interface configuration mode. (See Step 3.) If you configure the ISDN switch type for QSIG support using the **isdn switch-type** command in global configuration mode, you need not use the **isdn switch-type** command in interface configuration mode.
Step 2	`Router(config)# interface serial 1:`*channelnumber*	Enters interface configuration mode for the ISDN PRI interface. The *channelnumber* argument specifies the channel number. For T1, enter the channel number as **23**. For E1, enter **15**.
Step 3	`Router(config-if)# isdn switch-type primary-qsig`	(Optional) For the selected interface, configures the ISDN switch type to support QSIG signaling. Use this command if you did not configure the ISDN switch type for QSIG support globally in Step 1. **Note** For the selected interface, this command, entered in interface configuration mode, overrides any setting made with the **isdn switch-type** command entered in global configuration mode.
Step 4	`Router(config-if)# isdn overlap-receiving [T302` *value*`]`	Activates overlap signaling to send to the destination PBX. The keyword and argument are as follows: • **T302**—Specifies timer T302. • *value*—Specifies the value of timer T302 in milliseconds. The range is 500 to 20000.
Step 5	`Router(config-if)# isdn network-failure-cause [`*value*`]`	Specifies the cause code to pass to the PBX when a call cannot be placed or completed because of internal network failures. The *value* argument is a cause code from 1 to 127.

	Command	Purpose
Step 6	`Router(config-if)# isdn bchan-number-order {ascending | descending}`	(Optional) Configures the ISDN PRI interface to make the outgoing call selection in ascending or descending order. The keywords are as follows: • **ascending**—Makes the outgoing call selection in ascending order. • **descending**—Makes the outgoing call selection in descending order. This is the default. For descending order, the first call from the Cisco MC3810 uses channel 23 (T1) or channel 31 (E1). The second call then uses channel 22 (T1) or channel 30 (E1), and so on, in descending order. For ascending order, if the PRI group starts with 1, the first call uses channel 1, the second call uses channel 2, and so on, in ascending order. If the PRI group starts with a different time slot, the ascending order starts with the lowest time slot.
Step 7	`Router(config-if)# exit`	Exits interface configuration mode.
Step 8	`Router(config)# controller {T1 | E1} 1`	Enters controller configuration mode. QSIG is supported only on controller 1.
Step 9	`Router(config-controller)# pri-group timeslot [1-31]`	Configures the PRI group for either T1 or E1 to carry voice traffic. For T1, available time slots are 1–23, and for E1 available time slots are 1–31. You can configure the PRI group to include all the time slots available, or you can configure a select group of time slots for the PRI group. For example, if only time slots 1–10 are in the PRI group, enter the **pri-group timeslot 1-10** command. If the PRI group includes all channels available for T1, enter the **pri-group timeslot 1-24** command. If the PRI group includes all channels available for E1, enter the **pri-group timeslot 1-31** command. **Note** When a PRI group is configured, T1 time slot 24 or E1 time slot 16 is automatically assigned to handle D-channel signaling.

As shown in the procedure, you have a choice of configuring the **isdn-switch-type** command to support QSIG at either the global configuration level or at the interface configuration level. For example, if you have a QSIG connection on one line and on the BRI port, you can configure the ISDN switch type in one of the following combinations:

- Set the global **isdn-switch-type** command to support QSIG, and set the interface **isdn-switch-type** command for **interface bri 0** to a BRI setting such as 5ess.
- Set the global **isdn-switch-type** command to support BRI 5ess, and set the interface **isdn-switch-type** command for **interface serial 1:23** to support QSIG.
- Configure the global **isdn-switch-type** command to another setting (such as switch type VN3), and then set the interface **isdn-switch-type** command for **interface bri 0** to a BRI setting, and set the interface **isdn-switch-type** command for **interface serial 1:23** to support QSIG.

Note The **codec** command must be configured before any calls can be placed over the connection to the PINX. The default codec type is G.729a.

When voice dial peers are configured for use with QSIG PRI, voice port 1/1 is used for all bearer channels.

Configuring T-CCS

This section describes transparent common channel signaling (T-CCS) for Cisco 2600 series, 3600 series, 7200 series, and 7500 series routers and Cisco MC3810 multiservice concentrators and includes the following sections:

T-CCS Overview

T-CCS provides a way to interconnect PBXs and key telephone systems (KTSs) when the PINX does not support QSIG or when the PINX uses a proprietary solution. The following Cisco hardware provides support for T-CCS:

- Digital T1/E1 packet voice trunk network modules on Cisco 2600 series and 3600 series routers
- Two-port T1/E1 digital voice port adapters for Cisco 7200 series and 7500 series routers
- Digital voice module (DVM) on Cisco MC3810 multiservice concentrators

T-CCS allows the connection of two PBXs with digital interfaces that use a proprietary or unsupported CCS protocol without the need for interpretation of CCS signaling for call processing. T1/E1 traffic is transported transparently through the data network, and the T-CCS feature preserves proprietary signaling. From the PBX standpoint, this is accomplished through a point-to-point connection. Calls from the PBXs are not routed, but follow a preconfigured route to the destination.

CCS differs from a related technology, channel-associated signaling (CAS), in that it uses a separate transmission channel to relay signaling and address information in embedded packets conforming to standards recommendations. Examples of CCS signaling include Q.931 on ISDN Primary Rate Interface (PRI) and QSIG protocol signaling for PINX devices.

CAS, which is older than CCS, has evolved over many years and is supported on many Cisco routers. CAS signals and the dual tone multifrequency (DTMF) (or dial pulse) digits that indicate the telephone number of the called party are sent within the actual voice band transmission channel. Digital signal processors (DSPs) in Cisco voice nodes monitor these channels, decode the status and address signaling, and report status and state changes for the telephone calls.

If you are configuring your Cisco platform to route signaling traffic for Voice over Frame Relay (VoFR) or Voice over ATM (VoATM), you can configure T-CCS using T-CCS frame forwarding.

If you are configuring your Cisco platform to route signaling traffic for VoIP, T-CCS is configured by routing traffic over a clear channel codec.

The configuration procedures are described in the section "T-CCS Configuration Task List."

The T-CCS feature provides the following benefits:

- Efficient and cost-effective services on permanent (virtual) circuits or leased lines.
- PBX feature transparency across a WAN, permitting PBX networks to provide advanced features, such as calling name and number display, camp-on/callback, network call forwarding, centralized attendant, and centralized message waiting.
- Compressed Voice over Frame Relay, ATM, and IP support for virtually any CCS-based PBX.
- Dynamic allocation of bandwidth to voice calls using voice activity detection (VAD).

T-CCS Limitations

The T-CCS feature has the following restrictions:

- The digital T1/E1 packet voice trunk network module can have one or two slots for voice/WAN interface cards (VWICs); VWICs supply one or two ports. Only the dual-mode (voice/WAN) multiflex trunk cards are supported in the digital E1 packet voice trunk network module, and not older VICs.
- Drop-and-insert capability is supported only between two ports on the same multiflex card.
- Digital E1 voice is not manageable through Simple Network Management Protocol (SNMP) using existing versions of Cisco Voice Manager.
- On the Cisco MC3810, when T-CCS frame forwarding is configured, the speed (clock rate) of serial interface 1 of the Cisco MC3810 is limited to a maximum of 192 kbps. This restriction assumes that the multiflex trunk module (MFT) is installed in slot 3 on the Cisco MC3810. If the MFT is not installed, then serial interface 1 does not operate, but T-CCS frame forwarding is supported on other interfaces.
- The T-CCS feature supports PVCs, not SVCs.
- Cross-connections imply fractional trunks.
- For Frame Forwarding, preconfigured interfaces can be serial 0, serial 1, or T1/E1 0.

Related Documents for T-CCS

The following documents provide additional information to help implement T-CCS:

- *Cisco IOS Wide-Area Networking Configuration Guide*
- *Cisco IOS Wide-Area Networking Command Reference*
- *Cisco IOS Voice, Video, and Fax Command Reference*
- *Cisco IOS Debug Command Reference*
- *Configuring Cisco MC3810 Series Concentrators to Use High-Performance Compression Modules*
- *Voice Port Enhancements in Cisco 2600 and 3600 Series Routers and MC3810 Series Concentrators*
- *Voice over Frame Relay Using FRF.11 and FRF.12 Configuration Updates*

For hardware information, including information about the high-performance compression module (HCM), see the *Cisco MC3810 Multiservice Concentrator Hardware Installation Guide*.

T-CCS Prerequisite Tasks

The following configuration tasks should be completed before you configure a router for T-CCS:

- Obtain T1 or E1 service from your service provider.
- Establish a working network.
- Complete your company's dial plan.
- Establish a working telephony network based on your company's dial plan. For information about helpful documents, see the section "Related Documents for T-CCS."
- Install required multiflex trunk modules and voice components:
  - Digital T1/E1 packet voice trunk network modules on Cisco 2600 series and 3600 series routers
  - Two-port T1/E1 digital voice port adapters for Cisco 7200 series and 7500 series routers
  - DVM on Cisco MC3810 concentrators to support digital cross-connect voice (channel bank functionality)
  - High-performance compression modules (HCM) to support voice compression. See the section "Related Documents for T-CCS."
- Configure voice card and controller settings.
- Configure serial and LAN interfaces.
- Configure voice ports.
- Configure voice dial peers.

T-CCS Configuration Task List

To configure a router for T-CCS, complete the tasks shown in the following sections:

- Configuring T-CCS Cross-Connect, page 680
- Configuring T-CCS Frame Forwarding, page 684
- Configuring T-CCS for a Clear-Channel Codec, page 686

Note Although not always explicitly shown in these procedures, T-CCS also requires you to configure voice ports and dial peers.

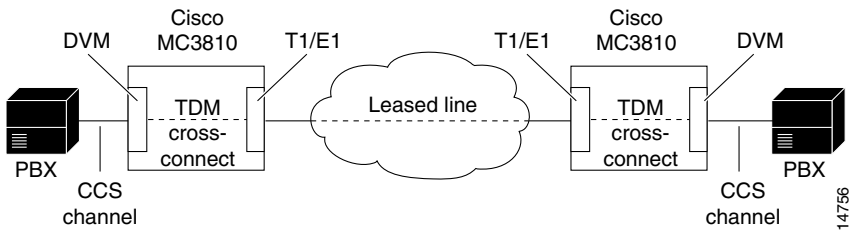
Configuring T-CCS Cross-Connect

This section is divided into the following procedures for T-CCS cross-connect:

- Configuring T1 and E1 TDM Groups, page 681
- Configuring T1 and E1 Trunk Bearer Channels, page 682

Figure 132 shows an example of T-CCS cross-connect. In this example, the CCS channel from the PBX is cross-connected on the Cisco MC3810 to a time slot on the T1/E1 controller. The channel is then passed through the WAN as a leased line to the second Cisco MC3810, where it is cross-connected to the DVM signaling time slot (time slot 24 for T1, or time slot 16 for E1). The channel is then passed to the second PBX. The CCS signal byte stream is passed through transparently by the Cisco MC3810.

Figure 132 T-CCS Cross-Connect Configuration



Configuring T1 and E1 TDM Groups

When you configure T-CCS cross-connect for E1 or T1, you set up time slot groups, and then configure cross-connect from the first T1/E1 controller to the second T1/E1 controller. The **mode ccs cross-connect** command allows the cross-connect. This command enables all the channels to perform similarly to normal CAS mode, except that the signaling bit is no longer processed by the router.

To configure T-CCS cross-connect, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} 0`	Enters controller configuration mode for controller E1 or T1 0.
Step 2	`Router(config-controller)# tdm-group` *tdm-group-no* `timeslots` *timeslot-list*	Configures a TDM channel group. The arguments are as follows: • *tdm-group-no*—Specifies a TDM channel group number. Valid values are from 0 through 23 for T1; from 0 through 30 for E1. • *timeslot-list*—Specifies a list of time slots. Valid values are from 1 to 24 for T1; from 1 to 15 and from 17 to 31 for E1. You can enter ranges or individual time slot numbers. **Note** Do not specify the **type** keyword in this command.
Step 3	`Router(config-controller)# exit`	Exits controller configuration mode for this controller.
Step 4	`Router(config)# controller {T1 | E1} 1`	Enters controller configuration mode for controller E1 or T1 1.
Step 5	`Router(config-controller)# mode ccs cross-connect`	Configures the controller to support CCS cross-connect and trigger the signaling channel.
Step 6	`Router(config-controller)# tdm-group` *tdm-group-no* `timeslots` *timeslot-list*	(T1 only) Configures a second TDM channel group. The arguments are as follows: • *tdm-group-no*—Specifies a TDM channel group number. Valid values are from 0 through 23 for T1; from 0 through 30 for E1. • *timeslot-list*—Specifies a list of time slots. Valid values are from 1 to 24 for T1; from 1 to 15 and from 17 to 31 for E1. You can enter ranges or individual time slot numbers. **Note** Do not specify the **type** keyword in this command.

	Command	Purpose
Step 7	`Router(config-controller)# ds0-group group-no timeslots timeslot-list type ext-sig`	(E1 only) Configures the specified channel group to support CCS mode. The keywords and arguments are as follows: • *group-no*—Specifies a TDM channel group number. Valid values are from 0 through 30. • **timeslots** *timeslot-list*—Specifies a list of time slots in the DS0 group. Valid values are from 1 to 31. You can enter ranges or individual time slot numbers. • **type**—The signaling method. • **ext-sig**—Specifies FRF.11 support. This keyword is available only when the **mode ccs cross-connect** command is enabled. **Note** The **ds0-group** command replaced the **voice-group** command that was supported in earlier releases. The **ext-sig** keyword replaced the **ext-sig-master** and **ext-sig-slave** keywords that were supported with the **voice-group** command.
Step 8	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 9	`Router(config)# cross-connect id-controller-1 tdm-group-no-1 controller-2 tdm-group-no-2`	Configures cross-connect pass-through between the two controllers. Depending on whether you are using T1 or E1, connect the two TDM controller groups (T1) or connect the TDM group controller to the DS0 group controller (E1).

Configuring T1 and E1 Trunk Bearer Channels

Tips After you configure a T-CCS connection by entering the **connection trunk** command, no change to the configuration takes place until the connection is shut down with a **shutdown** command and then restarted with a **no shutdown** command. For example, the phone number supplied in the **connection trunk** command can be changed while the connection is in the **no shutdown** state, but the change will not cause the current connection to be closed and a new connection to be opened to the new phone number. This will not take effect until the next **no shutdown** command following a **shutdown** command.

Note T-CCS cross-connect is not supported on analog PVC connections.

To use T-CCS cross-connect for bearer channels of the E1 or T1 trunk, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for the controller.
Step 2	`Router(config-controller)# mode ccs cross-connect`	Configures the controller to support CCS cross-connect. This command automatically creates serial interface 1:15 (E1) or 1:23 (T1).
Step 3	`Router(config-controller)# ds0-group group-no timeslots timeslot-list type ext-sig`	Configures the specified channel group to support CCS mode. The keywords and arguments are as follows: • *group-no*—Specifies a TDM channel group number. Valid values are from 0 through 23 for T1; from 0 through 30 for E1. • **timeslots** *timeslot-list*—Specifies a list of time slots. Valid values are from 1 to 24 for T1; from 1 to 31 for E1. You can enter ranges or individual time slot numbers. • **type**—The signaling method. • **ext-sig**—Specifies FRF.11 support. This keyword is available only when the **mode ccs cross-connect** command is enabled. **Note** The **ds0-group** command replaced the **voice-group** command that was supported in earlier releases. The **ext-sig** keyword replaced the **ext-sig-master** and **ext-sig-slave** keywords that were supported with the **voice-group** command.
Step 4	`Router(config-controller)# exit`	Exits controller configuration mode for this controller.
Step 5	`Router# (config)# voice-port slot/port`	Enters voice-port configuration mode. The arguments are as follows: • *slot*—Specifies the slot number. For digital voice ports, the *slot* number is 1 for this configuration. • *port*—Specifies the port number. Valid numbers are from 1 to 24 for T1 and from 1 to 15 or from 17 to 31 for E1. **Note** The *slot:port* format is also accepted.
Step 6	`Router# (config-voiceport)# connection trunk string`	Configures the voice-port connection. The *string* argument is the number of the voice channel that was configured as the **ext-sig** type for the **ds0-group** command.

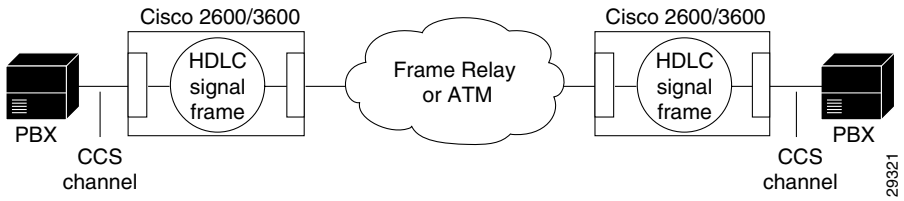
Configuring T-CCS Frame Forwarding

Cisco routers provide support for T-CCS frame forwarding, which allows a router to be connected to a Private Telco Network Exchange (PTNX) without having to interpret CCS signaling information for call processing. T-CCS frame forwarding forwards frames over a preconfigured interface running Frame Relay or ATM encapsulation.

With T-CCS frame forwarding, the connection between PTNXs over the network must be point-to-point and preconfigured. With the T-CCS frame forwarding implementation, calls from the PTNXs are not routed, but follow a preconfigured route to the destination.

Figure 133 shows an example of T-CCS frame forwarding. In the example, the first Cisco router captures the signaling frame from the PBX. The first Cisco router transports the signaling frame as a data frame through the Frame Relay or ATM network to the second Cisco router. The second Cisco router forwards the signaling frame to the PBX signaling channel.

Figure 133 T-CCS Frame Forwarding



To configure T-CCS frame forwarding, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} slot/port`	Enters controller configuration mode for the controller at the specified slot/port location. Valid values for slot and port are 0 and 1.
Step 2	`Router(config-controller)# mode ccs frame-forwarding`	Configures the controller to support CCS transparent signaling.

	Command	Purpose
Step 3	`Router(config-controller)# ds0-group` *ds0-group-no* `timeslots` *timeslot-list* `type ext-sig`	Defines the T1/E1 channels for use by compressed voice calls as well as the signaling method that the router uses to connect to the PBX or central office (CO). The keywords and arguments are as follows: • *ds0-group-no*—A value that identifies the DS0 group. Valid values are: 0 to 23 for T1; 0 to 30 for E1. • **timeslots** *timeslot-list*—Specifies a list of time slots in the DS0 group. Valid values are from 1 to 24 for T1; from 1 to 30 for E1. The value of *timeslot-list* can be a single number, a list of numbers separated by commas, or a pair of numbers separated by a hyphen to indicate a range of time slots. To map individual DS0 time slots, define additional groups. The router maps additional voice ports for each defined group. • **type**—The signaling method. • **ext-sig**—The signaling method selection for **type** depends on the connection that you are making: entering the keyword **ext-sig** specifies the external signaling interface, which signifies that the signaling traffic comes from an outside source. **Note** The **ds0-group** command automatically creates a logical voice port that is numbered as follows: *slot*/*port*:*ds0-group-no*. Although only one voice port is created, applicable calls are routed to any channel in the group. **Note** The **ds0-group** command replaced the **voice-group** command that was supported in earlier releases. The **ext-sig** keyword replaced the **ext-sig-master** and **ext-sig-slave** keywords that were supported with the **voice-group** command.
Step 4	`Router(config-controller)# no shutdown`	Activates the controller.
Step 5	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 6	`Router(config)# interface serial 1:`*channelnumber*	Enters interface configuration mode. This procedure maps the D channel from the digital T1/E1 packet voice trunk network module to the specified interface. The *channelnumber* argument specifies the channel number. For T1, enter the channel number as **23**. For E1, enter **15**.

	Command	Purpose
Step 7	`Router(config-if)# ccs encap frf11`	(Frame Relay only) Configures the CCS encapsulation to use the FRF11 packet format.
Step 8	`Router(config-if)# ccs encap atm`	(ATM only) Configures the CCS encapsulation to use the ATM packet format.
Step 9	`Router(config-if)# ccs connect {serial |atm} slot/number [dlci dlci| pvc vci | pvc vcd | pvc vpi/vci | pvc string]`	(Frame Relay and ATM) Configures the CCS connection. If the CCS connection is over Frame Relay, specify a serial interface and the DLCI. If the CCS connection is over ATM, specify ATM, slot and interface, and the PVC.
Step 10	`Router(config-if)# no cdp enable`	Disables Cisco Discovery Protocol (CDP) on the interface.
Step 11	`Router(config-if)# no keepalive`	Disables keepalive packets on the interface.

Configuring T-CCS for a Clear-Channel Codec

The T-CCS feature using a clear-channel codec allows tie-line emulation between two PBXs or PSTN switches running HDLC-based common channel signaling such as ISDN, DPNSS, CORNET, QSIG, and others. This configuration supports VoIP, VoFR and VoATM. Signaling frames are transparently forwarded on IP using an emulated 64-kbps channel. These frames travel over a clear-channel codec that is used on the voice port designated as the signaling channel. This codec passes data without changing the signaling frame.

T-CCS is configured when setting up the codec for the voice dial peer. The task table that follows sets up voice dial peers to support the local and remote stations. Not all possible commands are shown in the task table.

To learn more, see the *Cisco IOS Voice, Video, and Fax Command Reference*.

To configure T-CCS for a clear-channel codec, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} slot/port`	Enters controller configuration mode for the controller at the specified slot/port location. Valid values for slot and port are 0 and 1.
Step 2	`Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type ext-sig`	Defines the T1/E1 channels for use by compressed voice calls as well as the signaling method the router uses to connect to the PBX or CO. The keywords and arguments are as follows: • *ds0-group-no*—A value that identifies the DS0 group. Valid values are: 0 to 23 for T1; 0 to 30 for E1. • **timeslots** *timeslot-list*—Specifies a list of time slots in the DS0 group. Valid values are from 1 to 24 for T1; from 1 to 30 for E1. The value of *timeslot-list* can be a single number, a list of numbers separated by commas, or a pair of numbers separated by a hyphen to indicate a range of time slots. To map individual DS0 time slots, define additional groups. The router maps additional voice ports for each defined group. • **type**—The signaling method. • **ext-sig**—The signaling method selection for **type** depends on the connection that you are making: entering the keyword **ext-sig** specifies the external signaling interface, which signifies that the signaling traffic comes from an outside source. **Note** The **ds0-group** command automatically creates a logical voice port that is numbered as follows: *slot*/*port*:*ds0-group-no*. Although only one voice port is created, applicable calls are routed to any channel in the group. **Note** The **ds0-group** command replaced the **voice-group** command that was supported in earlier releases. The **ext-sig** keyword replaced the **ext-sig-master** and **ext-sig-slave** keywords that were supported with the **voice-group** command.
Step 3	`Router(config-controller)# no shutdown`	Activates the controller.
Step 4	`Router(config-controller)# exit`	Exits controller configuration mode.

	Command	Purpose
Step 5	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode and defines a local dial peer that will connect to the plain old telephone service (POTS) network. The keywords and arguments are as follows: • *number*—One or more digits identifying the dial peer. Valid entries are from 1 through 2147483647. • **pots**—Indicates a peer using a basic telephone service.
Step 6	`Router(config-dialpeer)# destination-pattern` *string* [`T`]	Configures the dial peer's destination pattern so that the system can reconcile dialed digits with a telephone number. The keywords and arguments are as follows: • *string*—A series of digits that specify the E.164 or private dialing plan phone number. Valid entries are the digits 0 through 9 and the letters A through D. The plus symbol (+) is not valid. You can enter the following special characters: – The star character (*) that appears on standard touch-tone dial pads can be in any dial string—but not as a leading character (for example, *650). – The period (.) acts as a wildcard character. – Use the comma (,) only in prefixes. The comma inserts a one-second pause. • **T**—(optional) The timer (T) character. When this character is included at the end of the destination pattern, the system collects dialed digits as they are entered—until the interdigit timer expires (10 seconds, by default) or the user dials the termination of end-of-dialing key (the default is #). **Note** The timer character must be a capital T.

	Command	Purpose
Step 7	`Router(config-dialpeer)# port slot/port:ds0-group-no`	Associates the dial peer with a specific logical interface. The arguments are as follows: • *slot*—Specifies the router location where the voice module is installed. Valid entries are from 0 through 3. • *port*—Specifies the voice interface card location. Valid entries are 0 and 1. • *ds0-group-no*—Indicates the defined DS0 group number. Each defined DS0 group number is represented on a separate voice port. This allows you to define individual DS0s on the digital T1 card.
Step 8	`Router(config-dialpeer)# exit`	Exits dial-peer configuration mode to complete the POTS dial-peer configuration.
Step 9	`Router(config)# dial-peer voice number voip`	Enters dial-peer configuration mode and defines a remote VoIP dial peer. The keywords and arguments are as follows: • *number*—One or more digits identifying the dial peer. Valid entries are from 1 through 2147483647. • **voip**—Indicates a VoIP peer using voice encapsulation on the IP network.
Step 10	`Router(config-dialpeer)# codec clear-channel`	Sets codec complexity to **clear-channel** to use the clear channel codec. **Note** The voice-card configuration **codec complexity** command sets the codec options that are available when you execute this command.
Step 11	`Router(config-dialpeer)# vad`	(Optional) Activates voice activity detection (VAD), which allows the system to reduce unnecessary voice transmissions caused by unfiltered background noise. **Note** This setting is enabled by default.

	Command	Purpose
Step 12	`Router(config-dialpeer)# destination-pattern` *string* `[T]`	Configures the dial peer's destination pattern so that the system can reconcile dialed digits with a telephone number. The keywords and arguments are as follows: • *string*—A series of digits that specify the E.164 or private dialing plan phone number. Valid entries are the digits 0 through 9 and the letters A through D. The plus symbol (+) is not valid. You can enter the following special characters: – The star character (*) that appears on standard touch-tone dial pads can be in any dial string—but not as a leading character (for example, *650). – The period (.) acts as a wildcard character. – Use the comma (,) only in prefixes. The comma inserts a one-second pause. • **T**—(optional) The timer (T) character. When this character is included at the end of the destination pattern, the system collects dialed digits as they are entered—until the interdigit timer expires (10 seconds, by default) or the user dials the termination of end-of-dialing key (the default is #). **Note** The timer character must be a capital T.
Step 13	`Router(config-dialpeer)# session target` {**ipv4:***destination-address* | **dns:**[**s.** | **d.** | **e.** | **u.**] *host-name*}	Configures the IP session target for the dial peer. The keywords and arguments are as follows: • **ipv4:***destination-address*—Indicates the IP address of the dial peer. • **dns:***host-name*—Indicates that the domain name server will resolve the name of the IP address. Valid entries for the argument are characters representing the name of the host device. There are also wildcards available for defining domain names with the keyword by using source, destination, and dialed information in the host name. For complete command syntax information, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.

Verifying the T-CCS Configuration

To verify the T-CCS configuration, perform the following steps:

Step 1 Enter the **show controllers e1** command (without specifying a slot and port number) to view the status for all controllers, or enter the **show controllers e1** command with a slot and port number to view the status for a particular controller. Make sure that the status indicates that the controller is up (line 2 in the following example) and no alarms (line 4 in the following example) or errors (lines 9, 10, and 11 in the following example) have been reported.

```
Router# show controllers e1 3/0

E1 3/0 is up.
  Applique type is Channelized E1 - balanced
  No alarms detected.
  alarm-trigger is not set
  Version info Firmware:19990702, FPGA:6
  Framing is CRC4, Line Code is HDB3, Clock Source is Line.
  Data in current interval (2 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail
Secs

!
controller E1 3/0
 mode ccs frame-forwarding
 channel-group 15 timeslots 16
 ds0-group 0 timeslots 15 type ext-sig
 ds0-group 1 timeslots 1 type ext-sig
 ds0-group 2 timeslots 2 type ext-sig
 ds0-group 3 timeslots 3 type ext-sig
 ds0-group 4 timeslots 4 type ext-sig
 ds0-group 5 timeslots 5 type ext-sig
 ds0-group 6 timeslots 6 type ext-sig
 ds0-group 7 timeslots 7 type ext-sig
 ds0-group 8 timeslots 8 type ext-sig
 ds0-group 9 timeslots 9 type ext-sig
 ds0-group 10 timeslots 10 type ext-sig
 ds0-group 11 timeslots 11 type ext-sig
 ds0-group 12 timeslots 12 type ext-sig
 ds0-group 13 timeslots 13 type ext-sig
 ds0-group 14 timeslots 14 type ext-sig
 ds0-group 16 timeslots 31 type ext-sig
 ds0-group 17 timeslots 17 type ext-sig
 ds0-group 18 timeslots 18 type ext-sig
 ds0-group 19 timeslots 19 type ext-sig
 ds0-group 20 timeslots 20 type ext-sig
 ds0-group 21 timeslots 21 type ext-sig
 ds0-group 22 timeslots 22 type ext-sig
 ds0-group 23 timeslots 23 type ext-sig
 ds0-group 24 timeslots 24 type ext-sig
 ds0-group 25 timeslots 25 type ext-sig
 ds0-group 26 timeslots 26 type ext-sig
 ds0-group 27 timeslots 27 type ext-sig
 ds0-group 28 timeslots 28 type ext-sig
 ds0-group 29 timeslots 29 type ext-sig
 ds0-group 30 timeslots 30 type ext-sig
```

Step 2 To display information about voice-port configuration, enter the **show voice port summary** command. The following example shows sample output:

```
Router# show voice port summary
                                        IN      OUT
PORT   CH SIG-TYPE   ADMIN OPER STATUS   STATUS   EC
====== == ========== ===== ==== ======== ======== ==
1:1    1     ext     up    up   on-hook  idle     y
1:2    2     ext     up    up   on-hook  idle     y
1:3    3     ext     up    up   on-hook  idle     y
1:4    4     ext     up    up   on-hook  idle     y
1:5    5     ext     up    up   on-hook  idle     y
1:6    6     ext     up    up   on-hook  idle     y
1:7    7     ext     up    up   on-hook  idle     y
1:8    8     ext     up    up   on-hook  idle     y
1:9    9     ext     up    up   on-hook  idle     y
1:10   10    ext     up    up   on-hook  idle     y
1:11   11    ext     up    up   on-hook  idle     y
1:12   12    ext     up    up   on-hook  idle     y
1:13   13    ext     up    up   on-hook  idle     y
1:14   14    ext     up    up   on-hook  idle     y
1:17   17    ext     up    up   on-hook  idle     y
1:18   18    ext     up    up   on-hook  idle     y
1:19   19    ext     up    up   on-hook  idle     y
1:20   20    ext     up    up   on-hook  idle     y
1:21   21    ext     up    up   on-hook  idle     y
1:22   22    ext     up    up   on-hook  idle     y
1:23   23    ext     up    up   on-hook  idle     y
1:24   24    ext     up    up   on-hook  idle     y
1:25   25    ext     up    up   on-hook  idle     y
1:26   26    ext     up    up   on-hook  idle     y
```

Step 3 To display information about voice calls, enter the **show voice call summary** privileged EXEC command. The following example shows sample output:

```
Router# show voice call summary

PORT      CODEC    VAD VTSP STATE            VPM STATE
========= ======== === ==================== ========================
1:1.1     g729ar8   y  S_CONNECT            S_TRUNKED
1:2.2     g729ar8   y  S_CONNECT            S_TRUNKED
1:3.3     g729ar8   y  S_CONNECT            S_TRUNKED
1:4.4     g729ar8   y  S_CONNECT            S_TRUNKED
1:5.5     g729ar8   y  S_CONNECT            S_TRUNKED
1:6.6     g729ar8   y  S_CONNECT            S_TRUNKED
1:7.7     g729ar8   y  S_CONNECT            S_TRUNKED
1:8.8     g729ar8   y  S_CONNECT            S_TRUNKED
1:9.9     g729ar8   y  S_CONNECT            S_TRUNKED
1:10.10   g729ar8   y  S_CONNECT            S_TRUNKED
1:11.11   g729ar8   y  S_CONNECT            S_TRUNKED
1:12.12   g729ar8   y  S_CONNECT            S_TRUNKED
1:13.13   g729ar8   y  S_CONNECT            S_TRUNKED
1:14.14   g729ar8   y  S_CONNECT            S_TRUNKED
1:17.17   g729ar8   y  S_CONNECT            S_TRUNKED
1:18.18   g729ar8   y  S_CONNECT            S_TRUNKED
1:19.19   g729ar8   y  S_CONNECT            S_TRUNKED
1:20.20   g729ar8   y  S_CONNECT            S_TRUNKED
1:21.21   g729ar8   y  S_CONNECT            S_TRUNKED
1:22.22   g729ar8   y  S_CONNECT            S_TRUNKED
1:23.23   g729ar8   y  S_CONNECT            S_TRUNKED
1:24.24   g729ar8   y  S_CONNECT            S_TRUNKED
1:25.25   g729ar8   y  S_CONNECT            S_TRUNKED
1:26.26   g729ar8   y  S_CONNECT            S_TRUNKED
```

Step 4 To display information about configured DS0 and TDM groups, enter the **show running-config** privileged EXEC command. The following example shows sample output:

```
Router# show running-config
.
.
.
controller T1 0
 tdm-group 1 timeslots 24
 framing esf
 linecode b8zs
 channel-group 0 timeslots 1-23 speed 64
!
controller E1 1
 mode ccs cross-connect
 tdm-group 1 timeslots 16
 clock source internal
 ds0-group 0 timeslots 1 type ext-sig
 ds0-group 2 timeslots 2 type ext-sig
 ds0-group 3 timeslots 3 type ext-sig
 ds0-group 4 timeslots 4 type ext-sig
 ds0-group 5 timeslots 5 type ext-sig
 ds0-group 6 timeslots 6 type ext-sig
 .
 .
 .
 ds0-group 23 timeslots 23 type ext-sig
 ds0-group 24 timeslots 24 type ext-sig
 ds0-group 25 timeslots 25 type ext-sig
 ds0-group 26 timeslots 26 type ext-sig
.
.
.
voice-port 1:0
 compand-type a-law
 timeouts wait-release 3
 connection trunk 3001
!
voice-port 1:2
 compand-type a-law
 timeouts wait-release 3
 connection trunk 3002
!
voice-port 1:3
 compand-type a-law
 timeouts wait-release 3
 connection trunk 3003
!
.
.
.
dial-peer voice 12 pots
 destination-pattern 4012
 port 1:12
!
dial-peer voice 13 pots
 destination-pattern 4013
 port 1:13
!
```

```
dial-peer voice 14 pots
 destination-pattern 4014
 port 1:14
!
!
cross-connect 1 E1 1 1 T1 0 1
```

Note For full configuration details, see the "T-CCS Configuration Examples" section on page 699.

Troubleshooting Tips for T-CCS

If the T-CCS connection does not come up, check for the following:

- Loose wires, splices, connectors, shorts, bridge taps, and grounds
- Backwards transmit and receive
- Mismatched framing types (for example, CRC-4 versus no-CRC-4)
- Transmit and receive pair separation (crosstalk)
- Faulty line cards or repeaters
- Noisy lines (for example, power and crosstalk)

If you see errors on the line or the line is going up and down, check for the following:

- Mismatched line codes (HDB3 vs. AMI)
- Improper receive level
- Frame slips due to poor clocking plan

Monitoring and Maintaining T-CCS and Frame Forwarding

To monitor your T-CCS configuration, use these commands as needed:

Command	Purpose
Router# **show frame-relay vofr** [*interface* [*dlci* [*cid*]]]	Displays information about FRF.11 subchannels and CIDs.
Router# **show interface serial0**	Displays information the serial interface used with VoFR, the DLCIs used on the interface, and the DLCI used for the Local Management Interface (LMI).
Router# **show frame-relay pvc** [interface *interface* [*dlci*] [*cid*]]	Displays information about Frame Relay PVCs, on all PVCs, or for a particular CID.
Router# **show atm pvc** [*vpi/vci*] [*name*]	Displays information about all configured ATM PVCs or about a particular PVC by virtual path identifier (VPI) and virtual channel identifier (VCI) numbers or by name.
Router# **show interface atm0**	Displays information about ATM interface configuration.

PBX Interconnectivity Configuration Examples

The following sections give sample configurations for both the QSIG and T-CCS PBX signaling formats.

QSIG Configuration Examples

This section contains two examples of QSIG configuration:

QSIG for VoIP Configuration Example

The following configuration example configures interface serial 1:23 for QSIG PRI and to act as the QSIG slave:

```
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname as5300A
!
ip subnet-zero
!
isdn switch-type primary-qsig
!
controller T1 0
 shutdown
!
controller T1 1
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2
 shutdown
!
controller T1 3
 shutdown
!
!
voice-port 1:D
!
!
dial-peer voice 3001 pots
 destination-pattern 3001
 port 1:D
!
dial-peer voice 4001 pots
 incoming called-number 4001
 direct-inward dial
!
dial-peer voice 4002 voip
 destination-pattern 4001
 session target ipv4:1.14.82.14
```

```
!
!
interface Ethernet0
 ip address 1.14.82.13 255.255.0.0
 no ip directed-broadcast
!
interface 1:23
 no ip address
 no ip directed broadcast
 isdn switch-type primary-qsig
 isdn protocol-emulate user
 isdn incoming-voice modem
!
interface FastEthernet0
 no ip address
 no ip directed-broadcast
 shutdown
!
ip default-gateway 1.14.0.1
ip classless
!
line con 0
 transport input none
line aux 0
line vty 0 4
 login
!
end

=====================================================
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname as5300B
!
ip subnet-zero
!
isdn switch-type primary-qsig
!
!
controller T1 0
 shutdown
!
controller T1 1
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 2
 shutdown
!
controller T1 3
 shutdown
!
!
voice-port 1:D
!
!
dial-peer voice 3001 pots
 incoming called-number 3001
```

```
 direct-inward-dial
!
dial-peer voice 3002 voip
 destination-pattern 3001
 session target ipv4:1.14.82.13
!
dial-peer voice 4001 pots
 destination-pattern 4001
 port 1:D
!
interface Ethernet0
 ip address 1.14.82.14 255.255.0.0
 no ip directed-broadcast
!
interface Serial1:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-qsig
 isdn protocol-emulate network
 isdn incoming-voice modem
!
interface FastEthernet0
 no ip address
 no ip directed-broadcast
 shutdown
!
ip default-gateway 1.14.0.1
ip classless
!
line con 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

QSIG PRI Signaling on the Cisco MC3810 Configuration Example

The following configuration example configures interface serial 1:15 for QSIG PRI and sets it to act as the QSIG master. The example shows other commands necessary for the configuration.

```
! version 12.2
 no service pad
 service timestamps debug uptime
 service timestamps log uptime
 no service password-encryption
 !
 hostname c3810a
 !
 network-clock base-rate 56k
 ip subnet-zero
 no ip domain-lookup
 ip host rb 10.1.1.1
 !
 isdn switch-type primary-qsig-master
 !
 !
 stun peer-name 10.1.1.1
 stun protocol-group 1 basic
 !
 controller E1 1
```

```
 clock source internal
 pri-group timeslots 1-2,16
!
!
!
interface Ethernet0
 ip address 144.254.156.169 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 no keepalive
!
interface Serial0
 ip address 10.1.1.2 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip route-cache
 no ip mroute-cache
 no arp frame-relay
 bandwidth 256
 no keepalive
 no fair-queue
 serial restart-delay 0
 frame-relay interface-dlci 30 voice-encap 80
 hold-queue 1024 out
!
interface Serial1
 no ip address
 no ip directed-broadcast
 encapsulation stun
 no ip route-cache
 no ip mroute-cache
 stun group 1
 stun route all interface Serial0 dlci 30
!
interface Serial1:15
 no ip address
 no ip directed-broadcast
 no logging event link-status
 isdn switch-type primary-qsig-master
 isdn bchan-number-order ascending
 no cdp enable
!
interface Switch0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no fair-queue
!
interface FR-ATM0
 no ip address
 no ip directed-broadcast
!
interface FR-ATM20
 no ip address
 no ip directed-broadcast
 no ip route-cache
 shutdown
!
router rip
 network 10.0.0.0
 network 144.254.0.0
!
ip classless
```

```
!
map-list atm1
!
map-class frame-relay A-relay
 frame-relay traffic-rate 256000 1540000
 no frame-relay adaptive-shaping
!
line con 0
 transport input none
line aux 0
line 2 3
line vty 0 4
 login
!
!
voice-port 1/1
!
voice-port 1/2
!
dial-peer voice 1 pots
 destination-pattern 2...
 port 1/1
!
dial-peer voice 3 pots
 destination-pattern 3
 port 1/3
!
dial-peer voice 5 pots
 destination-pattern 5
 port 1/5
!
dial-peer voice 6 pots
 destination-pattern 6
 port 1/6
!
dial-peer voice 10 vofr
 destination-pattern 4...
 session target Serial0 30
!
end
```

T-CCS Configuration Examples

This section contains two examples of T-CCS configuration:

- T-CCS over Frame Relay Configuration Example, page 699
- T-CCS over IP Configuration Example, page 701

T-CCS over Frame Relay Configuration Example

The following configuration example shows T-CCS frame forwarding on controller E1. Only relevant sections of the configuration are shown. The example assumes that the IP portion of the network is already in place.

```
hostname routerA
!
voice-card 1
!
controller E1 1/0
 mode ccs frame-forwarding
```

```
 channel-group 15 timeslots 16
 ds0-group 0 timeslots 15 type ext-sig
 ds0-group 1 timeslots 1 type ext-sig
.
.
.
 ds0-group 14 timeslots 14 type ext-sig
 ds0-group 17 timeslots 17 type ext-sig
.
.
.
 ds0-group 30 timeslots 30 type ext-sig
!
interface Serial0/0
 ip address 200.200.200.2 255.255.255.0
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 clockrate 2000000
 frame-relay traffic-shaping
 frame-relay class fr1
 frame-relay map ip 200.200.200.1 231 broadcast
 frame-relay interface-dlci 231
  vofr data 4 call-control 5
 frame-relay intf-type dce
!
```

The E1 interface must be set to **mode ccs frame-forwarding** to enable transparent forwarding of the HDLC signaling protocol through the DSP.

The **ds0-group** command links the specified time slot of the E1 interface to the corresponding voice port, which is automatically created by the router. This allows the voice port to be tied to the correspondent dial-peer using the connection trunk command. The **ext-sig** type specifies that the signaling traffic is coming from an external source.

The serial interface is set for frame relay traffic.

The example continues with the **voice-port** and **dial-peer** configuration.

```
voice-port 1/0:0
 compand-type a-law
 timeouts wait-release 3
 connection trunk 2000 answer-mode
.
.
.
voice-port 1/0:14
 compand-type a-law
 timeouts wait-release 3
 connection trunk 2014 answer-mode
!
voice-port 1/0:17
 compand-type a-law
 timeouts wait-release 3
 connection trunk 2017 answer-mode
.
.
.
voice-port 1/0:30
 compand-type a-law
 timeouts wait-release 3
 connection trunk 2030 answer-mode
!
```

```
dial-peer voice 2000 vofr
 destination-pattern 2000
 session target Serial0/0 231
!
dial-peer voice 1001 pots
 destination-pattern 1001
 port 1/0:1
.
.
.
dial-peer voice 1030 pots
 destination-pattern 1030
 port 1/0:30
!
```

The **dial-peer voice 2000 vofr** is used to forward the signaling channel over Frame Relay.

The **dial-peer pots** command sends the trunked voice DS0 traffic to the correspondant voice DS0 lines on the E1 port 1/0.

T-CCS over IP Configuration Example

The following configuration example configures T-CCS over IP using the clear-channel codec. The commands used in the configurations are explained inline. Only relevant sections of the configuration are shown. The example assumes that the IP portion of the network is already in place.

```
hostname routerA
!
voice-card 1
!
controller E1 1/0
 ds0-group 0 timeslots 16 type ext-sig
.
.
.
 ds0-group 10 timeslots 10 type ext-sig
!
interface Ethernet0/0
ip address 30.30.30.2 255.255.255.252
no ip directed-broadcast
!
voice-port 1/0:0
 compand-type a-law
 timeouts wait-release 3
 connection trunk 4000 answer-mode
!
voice-port 1/0:1
  compand-type a-law
  timeouts wait-release 3
  connection trunk 5001 answer-mode
.
.
.
voice-port 1/0:10
  compand-type a-law
  timeouts wait-release 3
  connection trunk 5010 answer-mode
!
```

The **ds0-group** command links the specified time slot of the E1 interface to the corresponding voice port, which is automatically created by the router. This allows the voice port to be tied to the corresponding dial peer using the connection trunk command. The **ext-sig** type specifies that the signaling traffic is coming from an external source.

The DS0 group assigned for signaling, configured as **ds0-group 0 timeslots 16**, must have the corresponding voice port and dial peer set for the clear-channel codec in order to enable transparent forwarding of the HDLC signaling protocol through the DSP.

The signaling DS0 channel of the E1 port 1/0 is configured to the dial peer whose destination pattern matches the number 4000. The **dial-peer voice 4000 voip** command is used to forward the signaling channel over IP.

The voice DS0 channels of the E1 port 1/0 are configured to the dial peer whose destination pattern matches the number 5... . The **dial-peer voice 5... voip** command is used to trunk the voice channels between routers.

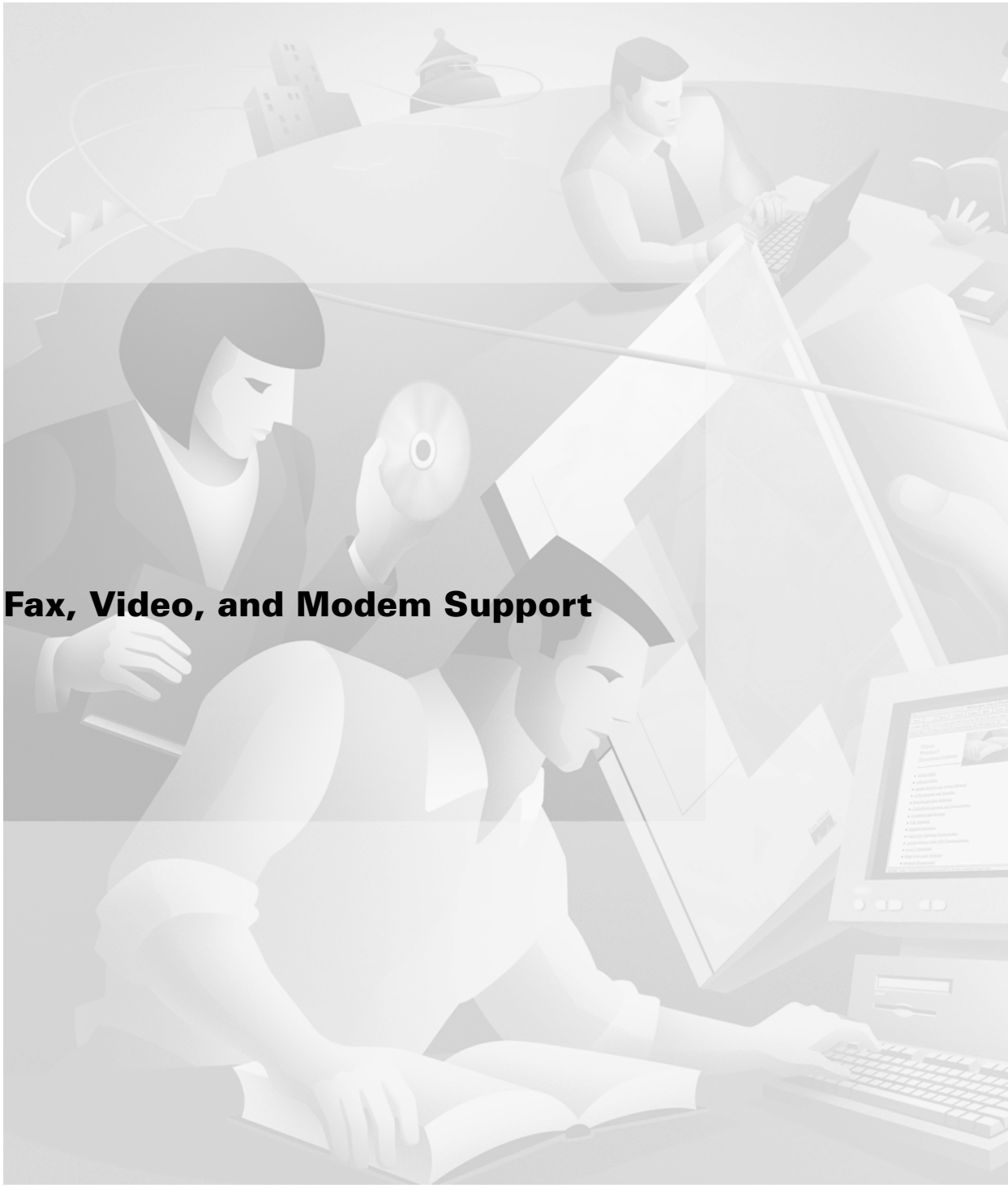
```
dial-peer voice 4000 voip
 destination-pattern 4000
 codec clear-channel
 session target ipv4:10.49.80.204
!
 dial-peer voice 3000 pots
 destination-pattern 3000
 port 2/0:0
!
dial-peer voice 5000 voip
 destination-pattern 5...
 session target ipv4:10.49.80.204
!
dial-peer voice 2001 pots
 destination-pattern 2001
 port 2/0:1
.
.
.
dial-peer voice 2010 pots
 destination-pattern 2010
 port 2/0:10
```

The **dial-peer voice 4000 voip** command is used to forward the signaling channel from the router over IP. The clear-channel codec must be applied to this dial peer in order to avoid that compression, and VAD will be applied to the signaling channel, which requires a transparent 644-kbps path through the DSP and the IP cloud.

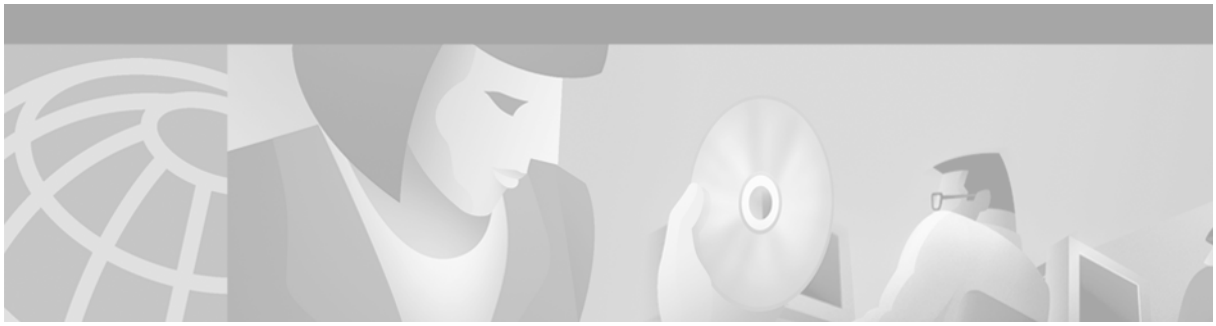
The **dial-peer voice 3000 pots** command forwards the incoming clear-channel signaling data to the corresponding signaling DS0 channel on the E1 port 1/0 of the router. This is achieved leveraging on the voice-port 1/0:0 created with **ds0-group 0 timeslots 16 type ext-sig**.

The **dial-peer voice 5000 voip** command is used to trunk the voice channels between routers. In this case, the codec used is the default G.729.

The **dial-peer voice 2001 pots** through **dial-peer voice 2010 pots** commands associate the VoIP legs of the trunked voice DS0s to the corresponding voice DS0s on the E1 port 1/0 of the router.



Fax, Video, and Modem Support



Configuring Fax Applications

This chapter describes T.37 Store and Forward Fax and T.38 Fax Gateway concepts and describes how to configure the fax applications for Cisco AS5300 universal access server access servers. The applications are T.37 Store and Forward Fax, T.38 Fax Relay for Voice over IP (VoIP) H.323, Fax Relay Packet Loss Concealment, and T.37/T.38 Fax Gateways. The applications enable the Cisco AS5300 universal access server to send and receive faxes across packet-based networks, using modems or voice feature cards (VFCs).

This chapter includes the following sections:

For a complete description of the commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information mentioned in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Fax Applications Overview

Fax applications enable Cisco AS5300 universal access servers to send and receive faxes across packet-based networks using modems or VFCs. Some of the benefits of the Fax Gateway are as follows:

- Universal inbox for fax and e-mail—Faxes and e-mails can go to the same mailbox using direct inward dialing (DID) numbers. E-mail and fax recipients can be combined.
- Toll bypass—In an enterprise environment in which offices in different cities are connected using a WAN, toll charges can be bypassed by transmitting faxes over the network connection. Because a fax message is stored on the mail server until Simple Mail Transfer Protocol (SMTP) forwards messages to the recipient, SMTP can forward fax e-mail attachments during off-peak hours (for example, during evenings and weekends), thereby reducing long-distance charges.
- Broadcast to multiple recipients—E-mail fax attachments can be sent to multiple recipients simultaneously.

- Improve robustness—The Fax Relay Packet Loss Concealment feature improves the robustness of the facsimile relay. It eliminates fax failures and lost data caused by excessive page errors. Field diagnostics and troubleshooting capabilities are improved by available debug commands. Statistics give better visibility into the real-time fax operation in the gateway, allowing for improved field diagnostics and troubleshooting.
- Cost savings and port density using T.37/T.38 Fax Gateway—The cost of maintaining one architecture (either fax or voice) is eliminated. Service providers can do the following:
  - Use a single port for voice, fax relay, and Store and Forward Fax. For smaller points of presence (POPs), the single-port configuration for these technologies is even more significant because mixed traffic can be handled more efficiently requiring only a single pool of ports versus splitting traffic across two pools.
  - Offer the new service of a single number for subscriber voice and fax access. The applications that use a single number for voice and fax require only half as many dialed number identification service (DNIS) numbers and dial peers as would be required with separate voice and fax applications.
  - Offer applications that require toggling from voice to fax. Applications such as never-busy fax service can be addressed once the gateway can dynamically switch from fax relay to fax store and forward.
- Interoperability with T.37 fax relay for VoIP H.323—The Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrator gateways with International Telecommunication Union Telecommunication (ITU-T) T.38 fax relay capability can interoperate with third-party gateways and gatekeepers over an IP H.323 network. The goal is to work with third-party gateways and gatekeepers to provide ITU-T standards-based T.38 fax relay services for multivendor networks.

The Cisco 2600 and 3600 series routers and Cisco MC3810 multiservice concentrator gateways provide standards-based toll bypass for fax and voice calls. In addition to existing voice and fax toll bypass capabilities, the multiservice gateways provide toll bypass for fax relay with the standards-based ITU-T T.38 fax relay implementation.

On-Ramp Gateway

The Cisco AS5300 universal access server acts as an on-ramp gateway to receive faxes from end users and uses call discrimination to determine call type and destination. It converts the faxes into TIFF files, creates standard Multipurpose Internet Mail Extension (MIME) e-mail messages, attaches the TIFF files to e-mail messages, and forwards the fax-mail messages to the messaging infrastructure of a designated SMTP server, where fax-mail messages are stored.

The on-ramp gateway uses the sending Message Transfer Agent (MTA) and dial peers to receive the faxes. The sending MTA, the Cisco AS5300 universal access server, defines delivery parameters associated with the e-mail message to which the fax TIFF file is attached. These delivery parameters include defining a return e-mail path or designating a destination mail server.

The on-ramp plain old telephone service (POTS) dial peers define the call as a fax transmission and identify the DNIS of the incoming fax call. The on-ramp Multimedia Mail over IP (MMoIP) dial peer defines the destination fax telephone number and the session target, which in this case is the SMTP server.

The configuration of the on-ramp gateway involves the following:

- Called subscriber number—Displayed number in the liquid crystal display (LCD) of the fax device sending a fax to a recipient. With a standard Group 3 fax device, this is the telephone number associated with the receiving fax device.
- Sending MTA—Contains the following elements in the e-mail message to which the fax TIFF file is attached:
  - Subject
  - Destination
  - Return path
  - Postmaster
  - Any additional identifying e-mail header information
  - Address to which any disposition notices are sent
- POTS dial peer—Defines the characteristics of the Public Switched Telephone Network (PSTN) connection between the sending fax device and the on-ramp gateway. The on-ramp gateway uses these characteristics to determine the call type and call destination using call discrimination.
- MMoIP dial peer—Describes the line characteristics generally associated with a packet network connection. With T.37 Store and Forward Fax, this is the IP network connection between the on-ramp gateway and the SMTP server. On-ramp MMoIP dial peers do the following:
  - Define the destination fax telephone number
  - Specify a destination e-mail address, which identifies the SMTP server
  - Define the image encoding and resolution specifics for the associated fax-mail TIFF files
  - Request DSNs, MDNs, or both.

    If DID is enabled, the incoming called number for the on-ramp POTS dial peer should match the destination pattern of the on-ramp MMoIP dial peer. If DID is not enabled, a redialer must be configured and enabled. In this case, the destination pattern must match the forwarded dialed digits from the redialer.

Off-Ramp Gateway

Off-ramp faxing requires that the Cisco AS5300 universal access server act as an off-ramp gateway and dial POTS and communicate with a remote Group 3 fax device using standard fax protocols. It uses call discrimination to determine call type and destination.

Off-ramp faxing activities are not mutually exclusive. An e-mail can be sent as a fax, and a TIFF file can be attached to it. When the Cisco AS5300 universal access server converts the e-mail to fax format, it also converts the attached TIFF file to standard Group 3 fax format.

The off-ramp gateway does the following:

- Converts a fax-mail TIFF file or plain text file into a standard format and delivers it to the recipient. Store and Forward Fax does not alter the TIFF or plain text file in any way from its original format when converting it into a standard fax format. The off-ramp gateway uses the receiving MTA and dial peers to perform the conversion.
- Delivers an e-mail message as a standard fax transmission. The Cisco AS5300 universal access server generates information that is appended to the top of each faxed page (text-to-fax pages) and creates a fax cover sheet. The off-ramp gateway uses the receiving MTA and dial peers to deliver e-mail messages as fax transmissions.

- Uses only POTS dial peers to define the line characteristics between the forwarding off-ramp gateway and the fax device. The dial peers also define the telephone number of the destination fax device. Number expansion can be used because the destination pattern is defined. As an option, the MMoIP dial peers can be configured, but MMoIP dial peers has limited functionality. They only define fax compression schemes and resolution and is useful only if those parameters are to be altered for the received fax-mails.
- Defines the parameters associated with the AS5300 SMTP server using the receiving MTAs. The MTAs can be SMTP host aliases, which can be different from the normal Domain Name System (DNS) host names, or an internal Cisco IOS host name.

The configuration of the on-ramp gateway involves configuring the following:

- Transmitting subscriber number—Displayed number in the LCD of the receiving fax device. Typically, with a standard Group 3 fax device, this is the telephone number associated with the transmitting or sending fax device.
- Fax transmission speed—Transmission speed of the fax device; this should be set to the speed of the other devices, if possible. This functionality is particularly helpful if the off-ramp gateway is sending faxes into an area where the fax transmission speed is always negotiated down to a slower speed.
- Receiving MTA—Accepts incoming mail (from the Cisco AS5300 universal access server to the SMTP server) if the destination host name of the incoming mail matches one of the aliases configured by the **mta receive aliases** command.
- Off-ramp POTS dial peer—Defines the line characteristics between the off-ramp gateway forwarding the converted e-mail message and the receiving fax device.
- Off-ramp MMoIP dial peer—Specifies a particular resolution for the fax transmission or defines an encoding type, which is optional. If the MMoIP dial peer is configured, the incoming called number must match the destination pattern telephone number of the corresponding on-ramp POTS dial peer.
- Faxed header information—Information appended to the top of each cover and text page indicates the telephone number of the sending fax device, the date, and the time of transmission. The header information is required.
- Fax cover page—Captures information taken from the originating e-mail messages. The destination address of an e-mail message controls the generation of a cover page on a per-recipient basis.

Call Discrimination Process

When the on-ramp gateway receives a call, it immediately identifies whether the call is being delivered using a PRI or T1 channel associated signaling (CAS) interface. If the call is on a T1-CAS interface, the gateway checks the service type field of the CAS group configuration. If the service type of the CAS group is fax, the interface forwards the fax to the MMoIP dial peer. If the gateway determines that the call is on a PRI interface, then the on-ramp gateway looks at several POTS dial peer data fields to determine what kind of call it has received.

POTS Dial Peers

The on-ramp gateway looks at the incoming called number field of each POTS dial peer listed in the dial peer lookup table. It compares the number configured as the incoming called number to the number received and selects the first POTS dial peer whose data matches. If the on-ramp router does not find a match, it assumes that the incoming call is a data call and processes it accordingly.

If the on-ramp router does find a match, it will then look at the service type field of the POTS dial peer to determine whether this is a voice or fax call. If this call has been flagged as a voice call, the on-ramp gateway will process it appropriately as a voice call.

If the call has been flagged as a fax call, the on-ramp gateway checks to see whether DID has been enabled. If DID has been enabled, the gatway concludes that the telephone number it has received is the destination directory number (DN) and forwards the call to be matched with the appropriate on-ramp MMoIP dial peer.

If DID has not been enabled, the on-ramp gateway assumes that the telephone number it received is the access DN. In this case, the on-ramp gateway provides a secondary dial tone and collects another telephone number from the redialer at the other end of the connection that the gateway will use as the destination DN. After the gateway has received this number from the redialer, the number is forwarded and matched to the appropriate on-ramp MMoIP dial peer.

A redialer is an interface hardware device that connects a fax device to the PSTN network. The user enters the complete telephone number into the fax device and the attached redialer captures and stores those dialed digits. It dials the on-ramp Cisco AS5300 universal access server that provides a secondary dial tone. Use a redialer when one of the following is true:

- Provisioning a DID service is not possible.
- User information, such as a personal ID number (PIN) from the redialer, is required.
- T1-CAS is in use.

The redialer should be programmed to wait two seconds and then send the PIN with destination digits to the on-ramp gateway.

The fax protocol starts after 52 digits have been detected or the interdigit timeout has exceeded 5 seconds. If the **debug fax receive** command is enabled, the digits are displayed as received by the on-ramp gateway. If a dial peer is matched, the fax proceeds. If a dial peer is not matched, the fax fails.

By default, DID is disabled, which means that the on-ramp gateway assumes that the fax call was placed using a redialer. When the call arrives, the gateway collects digits until it can identify the destination. Once the destination is identified, the gateway forwards the call to the next call leg (MMoIP dial peer).

If DID is enabled, the on-ramp gateway uses the called number (DNIS) to find a dial peer for the outgoing call leg. DID enables the gateway to match the incoming called number with a dial peer and then directly place the outbound call. With DID, the server does not present a dial tone to the fax machine and does not collect digits. It forwards the call directly to the configured destination.

The off-ramp gateway looks at the destination-pattern field of each POTS dial peer listed in the dial peer lookup table. It compares the number configured as the destination pattern with the destination DN portion of the fax-mail address and selects the first match.

After the off-ramp gateway has identified the appropriate POTS dial peer, it matches call type information. If the call type is identified as fax, it forwards the fax-mail message to off-ramp services. If the off-ramp router does not find a match, the recipient identified by the given address is not accepted by the off-ramp router.

MMoIP Dial Peers

The MMoIP function in the call discrimination process determines the fax-mail destination, which is the off-ramp gateway over which the fax-mail is sent to the destination fax machine. The on-ramp gateway looks at the destination pattern field of each MMoIP dial peer listed in the dial peer lookup table. It compares the number configured as the destination pattern with the number received and selects the first MMoIP dial peer whose the data matches.

The on-ramp gateway then looks at the session target field for the selected MMoIP dial peer in order to identify the destination of the fax-mail message. This value could be a specific off-ramp gateway or, if the fax is being delivered as an e-mail message, an e-mail address for a specific mail server.

The resolution of a fax image can be increased or decreased using the MMoIP dial peer configuration. Pass-through is the default: the image is sent exactly as it is received. Depending on the capacity of the fax machines in the network, a different image encoding (compression) scheme could be required for the fax TIFF image. The encoding default is pass-through.

On-Ramp Gateway Security

On-ramp gateway security controls who can send fax messages to the network. It is facilitated by authentication, authorization, and accounting (AAA) security services using RADIUS or TACACS+ as the local security protocol. On-ramp gateway faxing is a client of the authentication server, whether it is RADIUS or TACACS+. User information is forwarded to the AAA interface, and the authentication request is forwarded to the security server.

Authentication must be completed before the first page of faxed material is accepted from the modem by the Fax Application Process (FAP). If a response is not received from the AAA server before the first page is received, the fax modem or voice feature card (VFC) disconnects the call.

The on-ramp gateway inserts whatever value was configured in the “X-account-ID” field of the e-mail header that is used for authentication and accounting by the on-ramp gateway.

Attribute-Value Pairs for AAA

RADIUS attributes define specific AAA elements in a user profile, which is stored on the RADIUS server. The Cisco implementation of RADIUS supports Internet Engineering Task Force (IETF) and vendor-proprietary attributes. IETF RADIUS attribute 26 enables vendors to support extended attributes not suitable for general use. The Cisco fax applications use the RADIUS implementation of vendor-specific options in the recommended format.

Table 53 lists the supported vendor-specific options (subtype numbers from 3 through 21) using IETF RADIUS attribute 26 and the Cisco vendor-ID company code of 9.

Table 53 Vendor-Specific RADIUS Attributes

Subtype Number	Attribute	Description
3	Cisco-Fax-Account-Id-Origin	Account ID origin as defined by the system administrator for the **mmoip aaa receive-id** or the **mmoip aaa send-id** command.
4	Cisco-Fax-Msg-Id=	Unique fax message identification number.
5	Cisco-Fax-Pages	Number of pages sent or received during a fax session including cover pages.
6	Cisco-Fax-Coverpage-Flag	True/false flag that indicates whether a cover page was generated. True means a cover page was generated and false means it was not.
7	Cisco-Fax-Modem-Time	Number of seconds it takes to send fax data (x) and to complete the entire fax session (y) in the form x/y. For example, 10/15 means that the transfer time took 10 seconds and the full fax session took a total of 15 seconds.
8	Cisco-Fax-Connect-Speed	Modem speed. Possible values are 1200, 4800, 9600, and 14400.
9	Cisco-Fax-Recipient-Count	Number of recipients. Until e-mail servers support session mode, the number should be 1.

Table 53 Vendor-Specific RADIUS Attributes (continued)

Subtype Number	Attribute	Description
10	Cisco-Fax-Process-Abort-Flag	True/false flag indicating that fax session was aborted or successful. True is aborted and false is processed.
11	Cisco-Fax-Dsn-Address	Address to which DSNs are sent.
12	Cisco-Fax-Dsn-Flag	True/false flag to indicate if DSN is enabled. True is enabled and false is disabled.
13	Cisco-Fax-Mdn-Address	Address to which MDNs are sent.
14	Cisco-Fax-Mdn-Flag	True/Flash flag to indicate if MDN is enabled. True is enabled and false is disabled.
15	Cisco-Fax-Auth-Status	Authentication status—successful, failed, bypassed, or unknown.
16	Cisco-Email-Server-Address	E-mail server IP address handling the on-ramp fax-mail message.
17	Cisco-Email-Server-Ack-Flag	Acknowledgement that the e-mail server accepted the message.
18	Cisco-Gateway-Id	Processing gateway name in this format: hostname.domain-name.
19	Cisco-Call-Type	Type of call activity: fax receive or fax send.
20	Cisco-Port-Used	Slot/port number used to send or receive.
21	Cisco-Abort-Cause	System component that signalled an abort.

Access Control Lists

Incoming Access Control Lists (ACLs) can be used on Ethernet or FastEthernet interfaces to filter SMTP fax traffic. It is recommended that ACLs be configured to restrict access to the SMTP port (port 25) to only trusted e-mail servers. Creating ACLs is beyond the scope of this document. For information, refer to the *Cisco IOS Security Configuration Guide*.

ESMTP Accounting Services

Accounting information can be collected about fax services in two ways:

- Using RADIUS accounting
- Collecting the accounting information using SMTP

The extended simple mail transfer protocol (ESMTP) accounting feature enables the collection of accounting information as part of the SMTP session. This functionality is activated through the use of an intelligent fax client or MTA. In ESMTP accounting, the off-ramp gateway acting as an ESMTP server advertises capabilities to the MTA, which is acting as an e-mail client.

One of the capabilities the off-ramp gateway advertises is “xaccounting,” which supports ESMTP accounting. If the MTA recognizes the xaccounting service extension, the MTA (acting as the client) accepts the ESMTP accounting information sent from the off-ramp gateway. If the MTA does not recognize the xaccounting service extension, it does not send the **xact** command to the off-ramp gateway. In that case, the off-ramp gateway does not respond with ESMTP accounting data.

To use SMTP to collect accounting data, the MTA must be configured to explicitly request accounting information as part of the e-mail session. The MTA must be able to do the following:

- Recognize the xaccounting service extension during the extended hello (ehlo) transaction
- Send the **xact** command to the off-ramp gateway to activate the ESMTP accounting feature

Message Delivery Notifications

Described in RFC 2298, an message delivery notification (MDN) is a message that is sent to the originator of an e-mail message indicating that the e-mail message was received. MDN elements must be configured for both the on-ramp and off-ramp gateways. MDN requests as part of the on-ramp MMoIP dial peer configuration must be enabled. For complete instructions on how to configure MDNs, see the "Configuring MDNs" section on page 740.

Delivery Status Notifications

Delivery status notifications (DSNs) are messages or responses that are automatically generated and sent to the sender or originator of an e-mail message by the SMTP server, notifying the sender of the status of the e-mail message. DSNs must be configured for both the on-ramp and off-ramp gateways.

Three different states can be reported back to the sender as follows:

- Delay—Delivery of the message was delayed.
- Success—Delivery of the message was successful.
- Failure—Message was undeliverable to the SMTP server.

Because the delivery states are not mutually exclusive, messages for all or any combination of these events can be generated.

DSN requests can be enabled as part of the on-ramp MMoIP dial peer configuration. For complete instructions on how to configure DSNs, refer to the "Configuring DSNs" section on page 741.

T.37 Store and Forward Fax

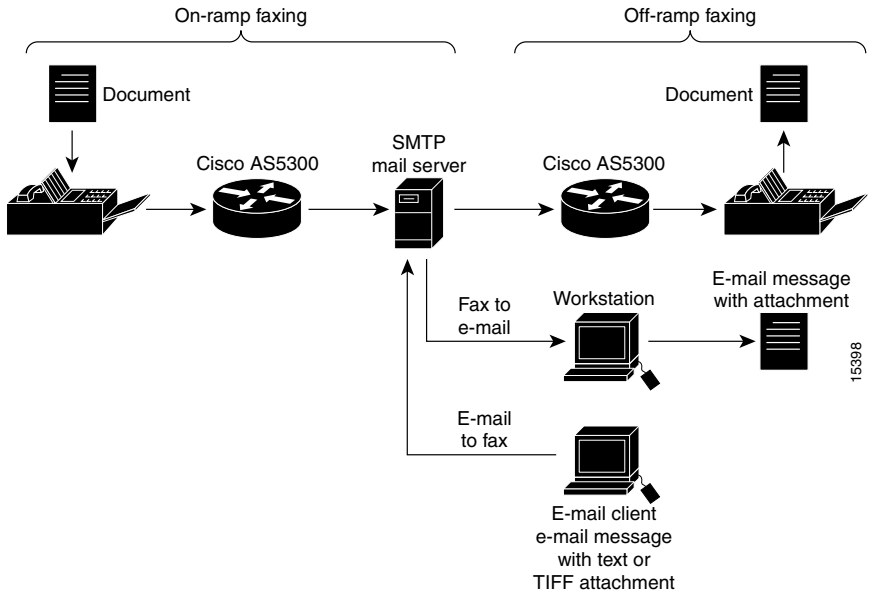
T.37 Store and Forward Fax is an implementation of the RFC 2305 proposed standard from the IETF and is the same as the T.37 recommendation of the International Telegraph Union (ITU). T.37 Store and Forward Fax enables the access server to become a multiservice platform, supplying both data and fax communication using modems.

T.37 Store and Forward Fax enables the following:

- Sending and receiving faxes to and from Group 3 fax devices
- Receiving faxes that are delivered as an e-mail attachment
- Creating and sending a standard e-mail message that is delivered as a fax to a standard Group 3 fax device

The basic functionality is facilitated through SMTP with additional functionality that provides confirmed delivery using existing SMTP mechanisms, such as ESMTP. Figure 134 shows a simple network topology using T.37 Store and Forward Fax.

Figure 134 T.37 Store and Forward Fax Functionality



The messaging infrastructure performs message routing, storage, and transport, and can be a standard Internet MTA—for example, UNIX sendmail or custom T.37 Store and Forward Fax software. The responsibility of delivering the fax-mail message falls to SMTP and the mail server.

Modem Pooling

As a default, T.37 Store and Forward Fax receives faxes on modems that are in the on-ramp gateway default modem pool. These modems are available for both fax and data calls. The on-ramp gateway determines the call type using DNIS and compares the DNIS number to the configured value for the incoming called-number POTS dial-peer configuration command.

If the DNIS number matches the incoming called number, DNIS treats the call as a fax transmission. If it does not find a match in its dial peer lookup table, it treats the call as a data call.

The incoming fax calls can be configured to bypass the default modem pool by defining a named modem pool. This is particularly useful if the calls have Modem ISDN channel aggregation (MICA) and Microcom faxes, because it diverts fax traffic from MICA modems that do not support fax transmission.

Fax Relay Packet Loss Concealment

Fax relay packet loss concealment improves the current real-time fax over IP (commonly known as fax relay) implementation in Cisco gateways, enabling fax transmissions to work reliably under higher packet loss conditions.

In addition, this feature includes enhanced real-time fax debug capabilities and statistics for improved field diagnostics and troubleshooting. The capabilities and statistics give better visibility into the real-time fax operation in the gateway.

One improvement is fax relay Error Correction Mode (ECM) on the VoIP dial peer. When used, the DSP fax relay firmware disables ECM through modification of the DIS T.30 message in both directions.

ECM provides for error-free page transmission. It is available on fax machines that include memory for storage of the page data (usually high-end fax machines). The page is transmitted in a series of blocks. After receiving the complete page data, the receiving fax indicates any frames with errors. The transmitting fax then retransmits those frames. This process is repeated until all frames have been received without errors. If the receiving fax is not able to receive an error-free page, the fax transmission may fail, and one of the fax machines may disconnect. With packet-loss levels greater than 2 percent, fax transmissions consistently fail between page transmissions when ECM is enabled.

When ECM is disabled, the page is sent using high-speed modulation in its raw encoded format. When detecting line errors with ECM disabled, the receiving fax has three options (in order of severity):

- Respond to page reception with the **ReTrain Positive** command. This causes the transmitting fax to go through the training check process before transmitting the next page.
- Respond to the page reception with the **ReTrain Negative** command. This causes the transmitting fax to go through the TCF process with a lower modulation scheme.
- Disconnect immediately.

Note ECM disable is recommended when there is a known lossy network (especially with packet loss at 2 percent or greater) and if fax traffic is anticipated for the dial peer.

Handling of Enclosures

All Cisco fax applications can process e-mail with the following MIME media content types:

- Text (plain type)
- Text (enriched type)
- Image or TIFF (“Profile S” described in RFC 2301)

Further, all Cisco fax applications support the following content transfer encodings:

- Seven bit
- Eight bit
- Base 64
- Quotable-printable

These content transfer encodings can be wrapped in any multipart/* content type. When messages with multiple sections are received, the first part of the multipart message is processed, and a count of what is and is not successfully sent is stored. The rest of the message is discarded. For example, if a multipart, alternative message has a plain text part and an enriched, html text part and the plain text is first, the the plain text part is the only part processed.

Note The TIFF file format must conform to RFC 2301 (*File Format for Internet Fax*). Store and forward fax does not support uuencoded text, JPEG or JBIG files, or multiraster content.

Caution The Cisco AS5300 universal access server recognizes only the listed file attachment types. If it receives a file format different from one of the defined acceptable formats, the data is discarded.

T.37/T.38 Fax Gateway

When the Cisco AS5300 universal access server is equipped with VFCs, it supports carrier-class Voice over IP (VoIP) and Fax over IP services. Since the Cisco AS5300 universal access server is H.323 compliant, it supports a family of industry-standard voice codecs and provides echo cancellation and voice activity detection (VAD)/silence suppression.

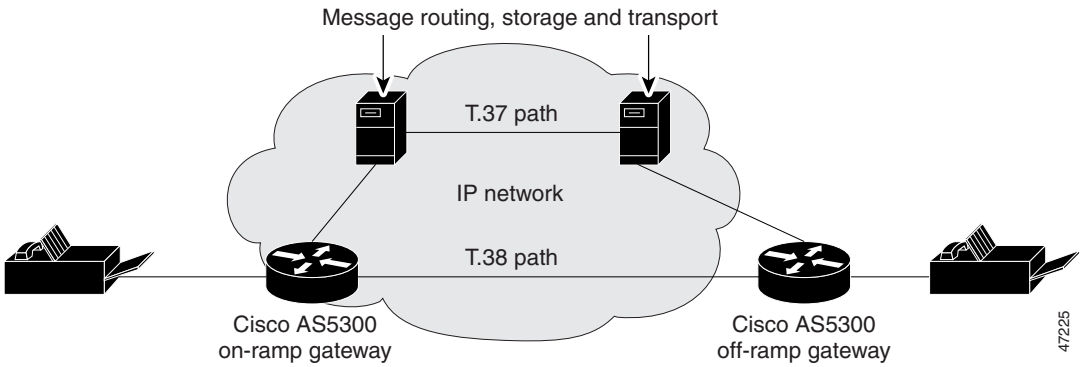
The VFC is a coprocessor card with a powerful reduced instructions set computing (RISC) engine and dedicated, high-performance DSPs to ensure predictable, real-time voice processing. The design enables steamlined packet forwarding. The Cisco AS5300 universal access server supports two VFCs that are scalable up to 96 E1 or 120 T1 voice connections within a single chassis.

T.37 Store and Forward Fax was supported by modem cards while the voice applications ran on the C542 digital signal processing module (DSPM) and C549 DSPMs that populated Cisco AS5300 VFCs. Each type of call required different technologies. With this software release, a single DSPM technology supports the following:

- Voice, fax relay, and T.37 Store and Forward Fax on both the C542 and C549 DSPM and the same voice port
- Dynamic switching from one application to another in the same call (IVR, voice, Fax Relay, and T.37 Store and Forward Fax)

Figure 135 highlights the real-time (T.38 path) versus the T.37 Store and Forward processing (T.37 path) for fax transactions over IP networks.

Figure 135 Real-Time Versus T.37 Store and Forward Fax Processing



Fax over IP used a proprietary protocol and an H.323 connection, represented by the T.37 path in the diagram. The T.37 path used the ESMTP T.37 Store and Forward method. The on-ramp gateway router accepted fax data from the PSTN fax machine.

The fax data was converted into a TIFF attachment in a MIME e-mail message and transmitted to a T.37 Store and Forward SMTP server. The server would deliver the fax-mail message to the off-ramp gateway. Once the off-ramp gateway received the fax-mail message, it processed the message and initiated a session with the destination fax machine.

With this software release, the T.38 path takes precedence over the T.37 path whenever possible. This means that as a fax session is being set up, the sending gateway first communicates using the T.38 path. If the communication fails, the sending gateway rolls over to the Cisco T.37 path if it is configured to rollover.

Using Interactive Voice Response

Interactive voice response (IVR) applications control calls by using voice prompts and digit collection in order to authenticate the user and identify the call destination. The applications are assigned to specific ports or invoked based on DNIS. They accommodate many gateway services by customizing the presentation of the interfaces to callers.

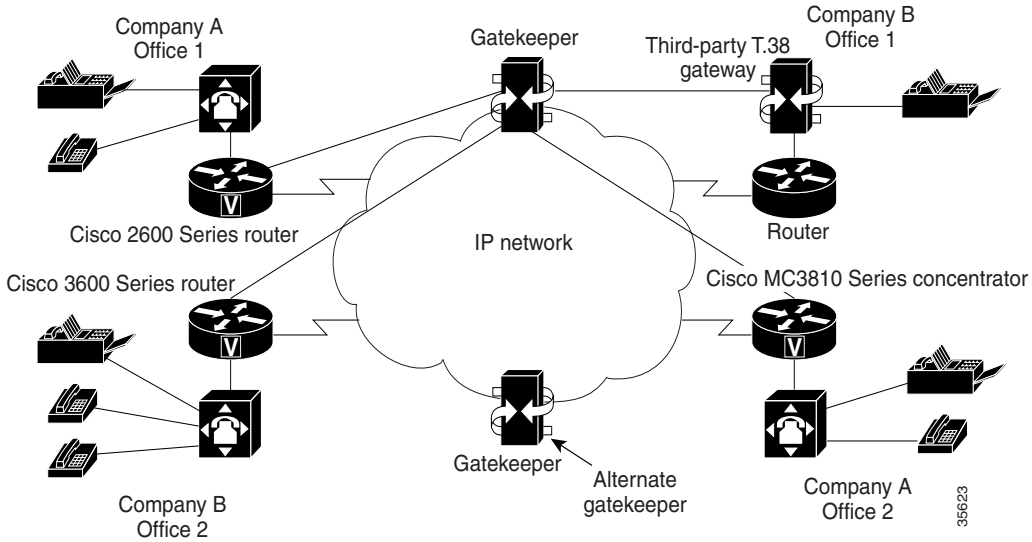
IVR uses Tool Command Language (TCL) scripts to gather information. For example, a TCL script plays when the caller receives a voice prompt to enter a specific type of information, such as a PIN. After the caller inputs the PIN, TCL collects the digits and forwards the digits to the server for storage and retrieval.

T.38 Fax Relay for VoIP H.323

The T.38 Fax Relay for VoIP H.323 feature provides standards-based fax relay protocol support on the Cisco 2600 and 3600 series routers and the Cisco MC3810 multiservice concentrator gateways. The Cisco proprietary fax relay solution is sometimes not an ideal solution for Enterprise and Service Provider customers who have implemented a mixed-vendor network. Because the T.38 fax relay protocol is standards based, Cisco gateways and gatekeepers can interoperate with third-party T.38-enabled gateways and gatekeepers in a mixed-vendor network when real-time fax relay capabilities are required.

 shows an IP H.323 network with Cisco and third-party gateways and gatekeepers using T.38 fax relay functionality. By using T.38 fax relay, all gateways and gatekeepers in this network are able to send faxes to other remote offices or to the offices of another company on the IP network.

Figure 136 IP Network for T.38 Fax Relay



For example, when a fax is sent from the originating gateway, a voice call is established. The terminating gateway detects the fax tone generated by the answering fax machine. The VoIP H.323 call stack then starts a T.38 mode request using H.245 procedures. If the opposite end of the call acknowledges the T.38 mode request, the initial audio channel is closed and a T.38 fax relay channel is opened. When the fax transmission is completed, the call reverts to voice mode.

Fax Applications Prerequisites

The following sections describe prerequisite tasks to perform before configuring all of the available fax applications:

Note If you are using modem cards, only T.37 Store and Forward Fax is supported. If you are using VFCs, T.37 Store and Forward Fax and T.38 Fax Relay and real-time fax are supported.

T.37 Store and Forward Fax Prerequisites

Before the T.37 Store And Forward Fax can be configured, the following tasks are required:

- Install a modem card into the appropriate slot of the Cisco AS5300 universal access server. Both MICA and Microcom modem cards support Store and Forward Fax, although MICA modem cards support only off-ramp faxing. For more information about installing Microcom and MICA modem cards, refer to the *Cisco AS5300 Universal Access Server Module Installation Guide* and the *Cisco AS5300 Universal Access Server Chassis Installation Guide*.
  - Update the Cisco AS5300 universal access server software configuration if modem cards are added or removed.
  - Download and install the V.90n firmware for the Microcom modem card and the standard portware with fax transmission capabilities for the MICA modem card.
- Establish a working IP network. For more information about configuring IP, refer to the "IP Overview," "Configuring IP Addressing," and "Configuring IP Services" chapters in the *Cisco IOS IP Routing Configuration Guide*.
- Complete the basic configuration for the Cisco AS5300 universal access server that includes, as a minimum, the following tasks:
  - Configure a host name and password for the Cisco AS5300 universal access server.
  - Configure the Ethernet 10Base T/100Base T interface so that the Cisco AS5300 universal access server can be recognized as a device on the Ethernet LAN.
  - Configure the Cisco AS5300 universal access server interfaces for ISDN PRI or T1 lines.
  - Configure the ISDN D channels for each ISDN PRI or T1 line.

  For more information about any of the these configuration tasks, refer to the *Cisco AS5300 Universal Access Server Software Configuration Guide*.

Note VoIP need not be configured for T.37 Store and Forward Fax to function.

The following sections describe specific prerequisite tasks to configure T.37 Store and Forward Fax:

Configuring the SMTP Server

Although it is not required, configuring the SMTP server enhances functionality. To configure the SMTP server, perform the following tasks:

- Edit the SMTP server alias file to include an alias for fax transmissions. The alias is an e-mail address that has the "fax=" prefix included in it. For example, fax=5551212, user@hostname.com. In this example, the on-ramp gateway automatically forwards the incoming fax to the mailbox for user@hostname.com.
  - If aliases are used to forward faxes, configure the on-ramp multimedia over IP (MMoIP) dial peer **session-target** command as **session target mailto: d@***hostname.com*. The **d** wildcard specifies that the destination fax machine telephone number is inserted in the to: field of the fax-mail that gets sent to the SMTP server.
- Modify parameters involving SMTP delivery requirements. Failure to do so can result in a monopoly of bandwidth and fax resources.

  Fax transmission has delivery requirements that are different from those of e-mail transmission. For example, in certain countries, it is illegal to try to send a fax more than three times in a row if transmission fails.

  SMTP mail delivery requirements are not governed by such strict regulations. In general, if an e-mail message cannot be delivered, the SMTP server is supposed to continue trying every 30 minutes for up to 5 days. To avoid any complications arising from the difference between the SMTP e-mail and fax delivery requirements, modify the following parameters:
  - Delivery to one recipient
  - Message priority
  - Connection cache size
  - Minimum queue age

Configuring the MTAs

MTAs, such as sendmail, Post.Office, and others, are normally configured to provide fast and reliable service for transferring e-mail. However, the needs of fax users are different. The best example of differing fax requirements is retry timeouts.

A typical MTA configuration will retry sending failed message transmissions every 30 minutes for up to 5 days. Resending e-mail every 30 minutes is usually unacceptable to fax users—they want retries more often than every 30 minutes and usually want transmission aborted well before the typical 5-day retry limit. Although a typical unmodified MTA can be used with the Cisco AS5300 universal access server for off-ramp operations, the MTA may need to be fine-tuned for fax operation.

Configuring Fax Operation

The Cisco AS5300 universal access server off-ramp accepts only one e-mail recipient per SMTP transaction because the SMTP server does not do the following:

- Queue messages in the Cisco AS5300 universal access server memory. The reason is the size of the messages and the lack of sufficient nonvolatile storage.
- Include a mechanism to enable the receiving MTA to indicate the success or failure of each delivery. It indicates the success or failure of the entire transaction.

The Cisco AS5300 universal access server prevents one SMTP transaction from going to multiple recipients by responding to the second and subsequent RCPT commands with a "450" reply code. Because of the typical mailer configuration, this causes a 30-minute delay for each recipient: immediate delivery for the first recipient, 30-minute delay for the second recipient, 60-minute delay for the third recipient, etc.

Configuring All Mail Through One Mailer

To simplify system administration, have all mail to the Cisco AS5300 universal access server go through one mailer by setting up a DNS MX record for the Cisco AS5300 universal access server. The record points to and sets up the mailer to skip MX record processing for the Cisco AS5300 universal access server. For example, the following two records would exist in DNS:

```
sj-offramp  in mx 10 sj-mailer
sj-offramp  in mx 20 sj-offramp
sj-offramp  in a  1.2.3.4
```

Configure ACLs to block incoming mail from other mailers. This prevents unauthorized use of the fax off-ramp and forces all mail to go through one mailer.

If ACLs have been set up on the router, the second MX record should *not* be placed in the DNS. For more information about ACLs, refer to the *Cisco IOS Security Configuration Guide*.

Configuring Sendmail 8.8.5 for Single Recipients

Fine-tuning sendmail 8.8.5 for a single recipient enables the Cisco AS5300 universal access server to work faster with Store and Forward Fax off-ramps and reduce delays caused by attempting to send to multiple recipients. It is important that sendmail be configured to send to each recipient serially, but without a delay after each transmission. Parallel configuration of sendmail with a single recipient and multiple sendmail client processes would cause a single message to be returned through sendmail, perhaps on a different port. The parallel configuration is not within the intended scope of this document.

Caution Do not modify the sendmail configuration on any system without a full understanding of what mail that system is processing and without the approval of the postmaster of the site. Modifying a company mail system can cause a loss of mail service upon which many companies rely for day-to-day operation.

To configure sendmail 8.8.5 to send to a single recipient, perform the following tasks:

- Modify the sendmail configuration file (usually named /etc/sendmail.cf) to the following:

```
Kmailertable hash /etc/mailertable
```

Note The line could already exist, but be commented out.

- If Kmailertable already exists in the configuration file, determine the name of the source text file used to build the mailertable.db file and edit it in or the existing mailertable.db of the site will be overwritten. If Kmailertable does not exist, add the line in toward the top of the configuration file with other "K" settings. The mailer table usually displays like this:

```
# not local -- try mailer table lookup
R$* <@ $+ > $*          $: < $2 > $1 < @ $2 > $3       extract host name
R< $+ . > $*            $: < $1 > $2                   strip trailing dot
R< $+ > $*              $: < $(mailertable $1 $) > $2  lookup
R< error : $- $+ > $*   $#error $@ $1 $: $2            check -- error?
R< $- : $+ > $*         $# $1 $@ $2 $: $3              check -- resolved?
R< $+ > $*              $: $>90 <$1> $2                try domain
```

Note A rewrite rule must be specified that causes a matching of the hosts in the mailer table. Ensure that the rewrite rules (starting with "R") for mailer table are not commented out.

  If the mailer table cannot be found, place the lines in Ruleset 0, which starts at the line containing "S0," before the rules that deliver local mail (R$=L $#local ...).

- Create a new mailer specification line in the section with other mailer specifications toward the bottom of the file as follows:

```
Mfaxofframp,    P=[IPC], F=DFMuXa0, S=11/31, R=21, E=\r\n, L=2040,
T=DNS/RFC822/SMTP,
A=IPC $h
```

  Ensure that the S and R values are the same as those for the existing mailer specifications for mail relaying. The existing S and R values are the lines beginning with an uppercase "M," usually toward the end of the sendmail.cf file.

  The S and R values control sendmail rewrite rules as applied to the Sender and Recipient addresses of the message. The rules and rule numbers must be different on each system, especially at sites that have complex sendmail configurations.

  It is important to omit the "F=m" flag and include the "F=0" (zero) flag as shown. The "m" flag causes delivery to multiple recipients (which is unwanted) and the "0" (zero) flag disables MX lookups (which are desired). The "0" (zero) flag is available only in sendmail version 8.8 or later. If an earlier version of sendmail is configured, omit the "0" and use [ ] in the mailer table.

- Create a file (/etc/mailertable.txt) with one line for each fax off-ramp device, listing the host name, white space, then the string "faxofframp:" and the host name again. For example, the hosts offramp-seattle.cisco.com and as5300-denver.cisco.com would be inputted as follows:

```
offramp-seattle.cisco.com  faxofframp:offramp-seattle.cisco.com
as5300-denver.cisco.com    faxofframp:as5300-denver.cisco.com
```

  If prior version of sendmail 8.8 is configured, use brackets around the right-side host name as follows:

```
offramp-seattle.cisco.com  faxofframp:[offramp-seattle.cisco.com]
```

- Input the following line to compile the new mailertable.txt using makemap (sometimes located in /usr/sbin):

```
/usr/sbin/makemap hash /etc/mailertable.db < mailertable.txt
```

Note If the system does not have makemap, sendmail will not support "hash." In this case, point sendmail at the mailertable.txt file by using "text" instead of "hash" on the Kmailertable line.

- Close and restart sendmail as follows:

```
ps -e | grep sendmail
kill pid                  # using PID indicated by above output
/usr/lib/sendmail -bd
```

- In DNS, set up A and MX records:

```
as5300-hostname     in a   a.b.c.d
in mx  10  sendmail-system
in mx  20  as5300-hostname
```

  This causes mail to be delivered to the sendmail-system first. Because the sendmail configuration disables MX lookups ("F=0") for the Cisco AS5300 universal access server, sendmail delivers directly to the IP address of the Cisco AS5300 universal access server. Also, if the sendmail system is down or otherwise unavailable, mail is queued directly to the Cisco AS5300 universal access server. Alternatively, use the following configuration:

```
as5300-hostname     in a   a.b.c.d
in mx  10  sendmail-system
in mx  20  backup-mta
```

  In this example, backup-mta is another sendmail (or other) mailer.

- Fine-tune the following parameters to control sendmail and provide near-real-time delivery of messages:
  - "O MinQueueAge" controls how long an entry must be in the queue before an attempt is made to process it. Reduce this setting for use with Cisco fax off-ramps (the normal value is 30 minutes).
  - "-q" switch starts sendmail and controls how often the queue is checked for reprocessing entries.
  - "O Timeout.queuereturn" controls the lifetime of a message in the queue.
  - "O Timeout.queuewarn" controls when sendmail issues a warning that the message has not been successfully relayed.
  - "O QueueSortOrder=XXX" controls how sendmail sorts the queue for processing. The string XXX should be one of these: host, priority, or time.
  - "O QueueLA" and "O QueueFactor" control the system load average, which causes sendmail to queue new messages instead of delivering them.
  - "O ConnectionCacheSize" and "O ConnectionCacheTimeout" processes more than one mail transaction in one TCP session with the fax off-ramp.
- Set the "O DoubleBounceAddress" parameter to the local postmaster or other administrative human address.

Note If the sending MTA supports the X-SESSION SMTP service extension, the Cisco AS5300 universal access server will support multiple recipients in one SMTP transaction and will store only one copy of each fax data page in its memory.

Configuring the Redialers

Perform the following tasks to enable a redialer:

- Program the redialer to dial the Cisco AS5300 universal access server acting as the on-ramp gateway and capture the dialed digits.
- Configure an MMoIP dial peer to match the forwarded dialed digits from the redialer.

Note Only the Mitel and Telecom Research redialers are supported on the Cisco AS5300 universal access server.

Fax Relay Packet Loss Concealment Prerequisite Tasks

VCWare 7.04 or higher version must be running before configuring fax relay packet loss concealment.

T.37/T.38 Fax Gateway Prerequisite Tasks

To enable the T.37/T.38 Fax Gateway for the Cisco AS5300 universal access server, perform the following tasks:

- Downloading VCWare to the VFC, page 722
- Copying Flash Files to the VFC, page 726
- Unbundling VCWare, page 727
- Adding Files to the Default File List, page 728
- Adding Codecs to the Capability List, page 728
- Deleting Files from VFC Flash Memory, page 729
- Erasing the VFC Flash Memory, page 729
- Configuring IVR, page 729

Downloading VCWare to the VFC

VFCs for the Cisco AS5300 universal access server come with a single bundled image of VCWare stored in VFC Flash memory. Table 54 shows the extension types defined for these embedded firmware files.

Table 54 VFC Firmware Extensions

Firmware	Filenames	Description
VCWare	vcw-vfc-*	Latest version of VCWare stored in Flash memory, including the following: • Datapath engine • Message dispatcher • DSP manager • VC manager • Process scheduler
DSPWare	btl-vfc-*	DSP bootloader
	cor-vfc-*	Core operating system and initialization
	bas-vfc-*	Base voice
	cdc-*-*	Voice codec files
	fax-vfc-*	Fax relay files

DSPWare is stored as a compressed file within VCWare. VCWare must be unbundled to install DSPWare in Flash memory. During the unbundling process, two default lists (default file and capability) are automatically created, populated with default files from that version of VCWare, and stored in VFC Flash memory. The default file list contains the names of the files that are initially loaded into DSP upon boot up, and the capability list defines the set of codecs that can be negotiated for a voice call.

VFC management enables the following functionality:

- Adding versions of VCWare to Flash memory by downloading and unbundling the files
- Erasing files contained in Flash memory
- Adding files to the default file and capability lists
- Deleting files from the default and capability lists

Before downloading VCWare to the VFC, determine whether or not the version of VFC ROM Monitor software is compatible with the installed Cisco IOS image. VFC ROM version 1.2 requires Cisco IOS image 0.14.1 (1.6 NA1) or later. VFC ROM Monitor version 1.2 can be made to work with Cisco IOS image 0.13 (or later) by appending the suffix ".VCW" to the VCWare image stored in VFC Flash memory.

The required tasks are as follows:

- Determining the Number of VFCs
- Identifying the VFC Mode
- Downloading the Software in VCWare Mode
- Downloading the Software in ROM Monitor Mode

Determining the Number of VFCs

To determine the number of installed VFCs and their location, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show vfc slot directory`	Determines the number of installed VFCs and their location.

For each VFC identified and located, upgrade the system software on that VFC.

Identifying the VFC Mode

To identify the mode (whether VCWare or ROM Monitor), use the following command in privileged EXEC mode:

Command	Purpose
`Router# show vfc slot board`	Determines whether the VFC is operating in VCWare mode or ROM Monitor mode.

If the mode is VCWare, the VFC status will be "VCWARE running." If the mode is ROM monitor, the VFC status will be "ROMMON."

Downloading the Software in VCWare Mode

To download VFC software to the VFC while in VCWare mode, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# erase vfc slot`	Erases the Flash memory.
Step 2	`Router# show vfc slot directory`	Displays that the VFC Flash memory is iempty.

Step 3	`Router# copy tftp: vfc:`
	or
	`Router# copy flash: vfc:`
		Downloads the VCWare from a TFTP Boot server into VFC Flash memory
		or
		Downloads the VCWare from the VFC motherboard into VFC Flash memory.
		Note The colons in this command are required.

Step 4	`Router# clear vfc slot`	Reboots the VFC.
Step 5	`Router# show vfc slot board`	Checks whether the VFC is back up in VCWare mode.
Step 6	`Router# show vfc slot directory`	Displays that VCWare is in the VFC Flash memory.
Step 7	`Router# unbundle vfc slot`	Unbundles the DSPWare from the VCWare and configures the default file list and the capability list.
Step 8	`Router# show vfc slot directory`	Displays that the DSPWare has been unbundled.

	Command	Purpose
Step 9	`Router# show vfc slot default-list`	Displays that the default file list has been populated.
Step 10	`Router# show vfc slot cap-list`	Displays that the capability list has been populated.

The Cisco AS5300 universal access server must be rebooted before these changes can take effect.

Note If the VFC ROM is version 1.1, the image name must end in ".VCW." If the VFC ROM is version 1.2, the image name must start with "vcv-."

Downloading the Software in ROM Monitor Mode

To download VFC software while in ROM monitor mode, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# clear vfc slot purge`	Erases the VFC Flash memory.

Step 2	`Router# copy tftp: vfc:`
	or
	`Router# copy flash: vfc:`
		Downloads the VCWare from a TFTP server into VFC Flash memory.
		or
		Downloads the VCWare from the VFC motherboard into VFC Flash memory.
		Note The colons in this command are required.

Step 3	`Router# clear vfc slot`	Reboots the VFC.
Step 4	`Router# show vfc slot board`	Checks whether the VFC is back up in VCWare mode.
Step 5	`Router# show vfc slot directory`	Displays that VCWare is in the VFC Flash memory.
Step 6	`Router# unbundle vfc slot`	Unbundles the DSPWare from the VCWare and configures the default file list and the capability list.
Step 7	`Router# show vfc slot directory`	Displays that the DSPWare has been unbundled.
Step 8	`Router# show vfc slot default-list`	Displays that the default file list has been populated.
Step 9	`Router# show vfc slot cap-list`	Displays that the capability list has been populated.

The Cisco AS5300 universal access server must be rebooted before these changes can take effect.

Note The image name must start with "vcw-."

Copying Flash Files to the VFC

Each VFC comes with a single bundled image of VCWare stored in Flash memory. VoIP for the Cisco AS5300 universal access server enables two different ways to copy new versions of VCWare to the VFC Flash memory by:

- Downloading from the Cisco AS5300 Motherboard, page 727
- Downloading from a TFTP Server, page 727

Downloading from the Cisco AS5300 Motherboard

To download from the AS5300 motherboard to Flash memory, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# copy flash: vfc:`	Downloads (copies) the Flash file from the Cisco AS5300 motherboard to the Flash memory on the VFC. **Note** The colons in this command are required.
Step 2	`Router# clear vfc slot`	Reboots the VFC.

Downloading from a TFTP Server

To download the latest version of VCWare from a TFTP server, ensure that the file is stored on the TFTP server. If a copy of the current version of VCWare is resident on disk, store that image on a TFTP server or the file cannot be downloaded into VFC memory. To copy the Flash file from a TFTP server, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# copy tftp: vfc:`	Downloads (copies) the Flash file from a TFTP server to the Flash memory on the VFC. **Note** The colons in this command are required.
Step 2	`Router# clear vfc slot`	Reboots the VFC.

Unbundling VCWare

VCWare must be unbundled before DSPWare can be loaded in Flash memory. The default file and capability lists are created and populated with the appropriate default files for that version of DSPWare. Table 55 shows the files associated with each firmware file.

Table 55 VFC Firmware Filenames

Firmware	Filenames
VCWare	vcw-vfc-mz.c542.t1.6
DSPWare Initialization and Static Files	btl-vfc-l.0.1.bin btj-vfc-l.0.1.bin jbc-vfc-1.3.0.bin cor-vfc-hc-1.3.4.24l.bin
DSPWare Overlay Files	bas-vfc-hc-1.3.4.24l.bin fax-vfc-hc-1.3.4.24l.bin cdc-g711-hc-1.3.4.24l.bin cdc-g726-hc-l.3.4.24l.bin cdc-g729-hc-1.3.4.24l.bin cdc-g728-hc-l.3.4.24l.bin cdc-g723.1-hc-l.3.4.24l.bin

To unbundle the current running image of VCWare, use the following command in privileged EXEC mode:

Command	Purpose
`Router# unbundle vfc slot`	Unbundles the current image of VCWare.

Adding Files to the Default File List

After the VCWare is unbundled, the default file list is automatically created and populated with the default files for VCWare. The default file list indicates which files are initially loaded into DSP at boot up. The following example shows the output from the **show vfc def** command, which displays the contents of the default file list:

```
Router# show vfc 1 def

Default List for VFC in slot 1:
1. btl-vfc-1.0.13.0.bin
2. cor-vfc-1.0.1.bin
3. bas-vfc-1.0.1.bin
4. cdc-g729-1.0.1.bin
5. fax-vfc-1.0.1.bin
6. jbc-vfc-1.0.13.0.bin
```

Under most circumstances, these default files should be sufficient. If needed, files from those stored in VFC Flash memory can be added to the default file list or existing files replaced from the default file list. When a specific file is added to the default file list, it replaces the existing file with the same extension type.

To add a file to the default file list, use the following command in global configuration mode:

Command	Purpose
`Router(config)# default-file filename vfc slot`	Selects a file stored in the Flash memory to be added to the default file list.

Adding Codecs to the Capability List

The capability list defines the set of codecs that can be negotiated for a voice call. Like the default file list, the capability list is created and populated when VCWare is unbundled and DSPWare added to VFC Flash memory. The following example shows the output from the **show vfc cap** command, which displays the contents of the capability list:

```
Router# show vfc 1 cap

Capability List for VFC in slot 1:
1. fax-vfc-1.0.1.bin
2. bas-vfc-1.0.1.bin
3. cdc-g729-1.0.1.bin
4. cdc-g711-1.0.1.bin
5. cdc-g726-1.0.1.bin
6. cdc-g728-1.0.1.bin
7. cdc-gsmfr-1.0.1.bin
```

Codec files can be added, using VFC management, if needed for a specific telephony network.

Note The capability list does not indicate codec preference: it only reports available codecs. The session application decides which codec to use.

To add a codec overlay file to the capability list, use the following command in global configuration mode:

Command	Purpose
`Router(config)# cap-list filename vfc slot-number`	Selects a codec overlay file to be added to the capability list.

Deleting Files from VFC Flash Memory

In some instances, a file may need to be deleted from the default file or capability lists. To delete a file from VFC Flash memory, use the following command in privileged EXEC mode:

Command	Purpose
`Router# delete file-name vfc slot`	Deletes the specified file from VFC Flash memory.

Erasing the VFC Flash Memory

When upgrading the Cisco AS5300 universal access to a more current version of VCWare, new files are stored in VFC Flash and do not overwrite existing files. The contents of VFC Flash memory must be erased to free memory space. To erase the Flash memory of a specific VFC, use the following command in privileged EXEC mode:

Command	Purpose
`Router# erase vfc slot`	Erases the Flash memory on the VFC.

Configuring IVR

Before configuring the Cisco gateways to support IVR, perform the following tasks:

- Configure VoIP to support H.323 compliant gateways, including specific devices in the network to act as gateways, such as configuring dial peers and voice ports.
- Configure a TFTP server to perform storage and retrieval of the required audio files.
- Download the appropriate TCL IVR script from the Cisco.com. Use the **copy** command to copy the audio file (.au file) to Flash memory, and the **audio-prompt load** command to read it into RAM. For more information about copying files into Flash memory, refer to "Copying Flash Files to the VFC" section on page 726.
- Ensure that the audio files are in the proper format. The IVR prompts require audio file (.au) format with 8-bit, u-law, and 8-Khz encoding. To encode the audio files, one of these two audio tools (or a equivalent tool) is recommended:
  - Cool Edit, manufactured by Syntrillium Software Corporation
  - AudioTool, manufactured by Sun Microsystems

- Ensure that the access platform has a minimum of 16 MB of Flash memory and 64 MB of DRAM.
- Install and configure the appropriate RADIUS security server in the network. The version of RADIUS must be able to support IETF-supported Vendor-Specific Attributes (VSAs), which are implemented by using IETF RADIUS Attribute 26.

T.38 Fax Relay for VoIP H.323 Prerequisites

Ensure that the following have been performed or checked before configuring VoIP H.323 for the T.38 fax relay:

- Cisco IOS Release 12.1(3)T is running on the Cisco AS5300 universal access server.
- There is a working VoIP H.323 network for voice calls.
- There has been complete voice interoperability testing with third-party gateways and gatekeepers.
- There is a minimum of 64 MB RAM.

Note Although 96 to 128 MB RAM is recommended, the memory requirement is dependent on the platform and the anticipated number of calls to be made through the system.

Fax Applications Configuration Tasks List

The configuration tasks for fax applications are described in the following sections:

Configuring the On-Ramp Gateway

To configure the on-ramp gateway, perform the tasks described in the following sections:

Configuring the Called Subscriber Number

To configure the called subscriber number, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# fax receive called-subscriber {$d$ | string}`	Defines the number that is displayed in the LCD of the sending fax machine. This parameter defines the called subscriber identification (CSI).

Configuring the Sending MTA

Defining the originator of the e-mail fax, the destination mail server, the subject of the message, and the postmaster, which is the default mail station for undeliverable e-mail message, is required (Steps 1 through 5). Steps 6 and 7 are optional.

Note The To: address of the fax-mail comes from the **session target** command configured for the MMoIP dial peer for the on-ramp gateway.

To configure the sending MTA, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mta send mail-from hostname string`	Specifies the originator host name of the e-mail fax message. Use this command with the **mta send mail-from username** command for a complete address.
Step 2	`Router(config)# mta send mail-from {username string | username $s$}`	Specifies the originator username of the e-mail fax message. Use this command with **mta send mail-from hostname** command for a complete address. The keyword **username s** is the calling number.
Step 3	`Router(config)# mta send server {host-name | IP-address}`	Specifies the destination server. **Note** DNS MX records are not used to determine the IP address of the host specified with the **mta send server** command.
Step 4	`Router(config)# mta send subject string`	Defines the text that appears in the subject field of the e-mail fax message.
Step 5	`Router(config)# mta send postmaster e-mail-address`	Defines sending address as the **mta send mail-from** address if the evaluated string is blank.
Step 6	`Router(config)# mta send origin-prefix string`	(Optional) Defines additional identifying information to be prepended to the e-mail header.
Step 7	`Router(config)# mta send return-receipt-to {hostname string username string}`	(Optional) Specifies the address where MDNs are sent, if MDNs are requested.

Configuring POTS Dial Peers

To configure the on-ramp gateway POTS dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `pots`	Defines the POTS dial peer tag number and enters dial-peer configuration mode.
Step 2	`Router(config-dial-peer)# application` *name*	Associates a specific IVR application with this dial peer. **Note** The **out-bound** keyword is not used with the POTS dial peers, but is used in the MMoIP dial peer configuration.
Step 3	`Router(config-dial-peer)# information-type fax`	Identifies calls associated with this dial peer as being fax transmissions, as opposed to being voice calls.
Step 4	`Router(config-dial-peer)# direct-inward-dial`	(Optional) Specifies DID. If a redialer is not used, DID must be enabled.
Step 5	`Router(config-dial-peer)# incoming called-number` *string*	Defines the telephone number associated with the POTS dial peer. If DID is enabled, the incoming called number (DNIS number) is used to match the destination pattern of outgoing MMoIP dial peers.
Step 6	`Router(config-dial-peer)# max-conn` *number*	(Optional) Defines the maximum number of on-ramp connections used simultaneously on this Cisco AS5300 to send fax-mail.

Note E.164 e-mail addresses that are compliant with RFC 2304 use this format: fax=+d@your.hostname.com format. If the off-ramp gateway receives the correct format, it strips the + and matches an off-ramp POTS dial peer with the remaining digits. The number contained in "d" must be a fully qualified E.164 telephone number (that is, it must include the country code) and it must not include an access code (such as "9" to get an outside line).

Configuring MMoIP Dial Peers

To configure the on-ramp gateway MMoIP dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `mmoip`	Defines the MMoIP dial peer tag number and enters dial-peer configuration mode.
Step 2	`Router(config-dial-peer)# application` *name* [`out-bound`]	Associates a specific IVR application with this dial peer. If the **out-bound** keyword is used, the named application handles the MMOIP dial peer in the outgoing mode.

	Command	Purpose
Step 3	`Router(config-dial-peer)# destination-pattern [+]string`	Identifies the destination fax telephone number. If DNIS has been enabled, this number should be the same as the configured incoming called number. If DNIS is not enabled, this should be the number from the redialer DNIS.
Step 4	`Router(config-dial-peer)# session target {mailto:{name | $d$}@domain-name | ipv4:destination-address | dns:[$s$. | $d$. | $u$. | $e$.] host-name| loopback:rtp |loopback:compressed | loopback:uncompressed}`	Defines the destination e-mail address for the fax-mail, meaning the e-mail address identifying the SMTP server.
Step 5	`Router(config-dial-peer)# session protocol smtp`	Identifies the session protocol being used between the on-ramp gateway and the remote mail server as SMTP.
Step 6	`Router(config-dial-peer)# image encoding {mh | mr | mmr | passthrough}`	Selects a specific encoding method for the fax-mail messages forwarded via this dial peer.
Step 7	`Router(config-dial-peer)# image resolution {fine | standard | super-fine | passthrough}`	Selects a specific resolution for the TIFF images attached to the fax-mail message forwarded by this dial peer.
Step 8	`Router(config-dial-peer)# max-conn number`	(Optional) Defines the maximum number of connections used simultaneously to send fax-mail.
Step 9	`Router(config-dial-peer)# dsn {delay | failure | success}`	(Optional) Requests that a delivery status notification be generated by the last-hop mailer if the delivery was successful. This DSN is sent to the address specified by the **mta send mail-from** command. Three types of DSNs can be requested: delay, failure, and success. **Note** DSN must be supported by the remote mail server.
Step 10	`Router(config-dial-peer)# mdn`	(Optional) Requests that a message disposition notification be generated by the mail user agent when the message is processed (typically opened or read). The MDN is generated by the receiving mail user agent and sent to the address defined by the **mta send return-receipt-to** command. **Note** Return receipt must be supporte or initiated by the receiving e-mail client.

Verifying the Gateway Configuration

To verify the gateway configuration, perform the following tasks:

- Verify the configured called-subscriber number using the **debug fax receive called-number** command.
- Check the configured called subscriber number by sending a fax and checking the number in the sending machine LCD.
- Verify that the dial peers have been configured correctly using the **show dialplan number fax** command.

- Display Class 2 fax tracing information on all on-ramp fax connections using the **debug fax receive all** command.
- Display output for all of on-ramp client connections (messages exchanged; for example, the handshake) between the e-mail server and the on-ramp gateway using the **debug mta send all** command.
- Display output for a specific on-ramp SMTP client connection during e-mail transmission using the **debug mta send rcpt-to** command.
- Test connectivity between the on-ramp gateway and the e-mail server by sending a test e-mail to a specified e-mail address and using the **debug mmoip send email** command.
- Make a POTS call to the on-ramp gateway and listen for a secondary dial tone to ascertain if DID is enabled or disabled.

Configuring the Off-Ramp Gateway

To configure the off-ramp gateway, perform the tasks in the following sections:

- Configuring the Transmitting Subscriber Number, page 734 (Required)
- Configuring the Fax Transmission Speed, page 734 (Required)
- Configuring the Receiving Mail Transfer Agent, page 735 (Required)
- Configuring POTS Dial Peers, page 732 (Required)
- Configuring MMoIP Dial Peers, page 732 (Required)
- Configuring the Faxed Header Information, page 736 (Required)
- Configuring the Fax Cover Page Information, page 737 (Required)

Configuring the Transmitting Subscriber Number

To configure the transmitting subscriber number, use the following command in global configuration mode:

Command	Purpose
`Router(config)# fax send transmitting-subscriber {$d$ | string}`	Defines the number that appears in the LCD of the receiving fax device. This parameter defines the transmitting subscriber identification (TSI).

Configuring the Fax Transmission Speed

To configure the fax transmission speed, use the following command in global configuration mode:

Command	Purpose
`Router(config)# fax send max-speed {12000 | 14400 | 2400 | 4800 | 7200 | 7600}`	Specifies the maximum fax speed.

Configuring the Receiving Mail Transfer Agent

To configure the receiving MTA, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mta receive aliases string`	Defines a host name to be used as an alias for the off-ramp Cisco AS5300 universal access server device. Up to ten different aliases can be specified. The Cisco AS5300 universal access server SMTP server accepts only incoming mail if the destination host name of the incoming mail matches one of the aliases as configured by the **mta receive aliases** command. A domain IP address must be explicitly added by enclosing the address in brackets (for example, [xxx.xxx.xxx.xxx]).
Step 2	`Router(config)# mta receive generate-mdn`	(Optional) Configures the Cisco AS5300 universal access server to actually generate an MDN message when requested to do so. Some sites may want to enable or disable this feature depending on the types of mailers in use.
Step 3	`Router(config)# mta receive maximum-recipients number`	Defines the number of simultaneous SMTP recipients handled by this device. This is intended to limit the number of resources (modems) allocated for fax transmissions.

Configuring the POTS Dial Peer

To configure the POTS dial peer for the off-ramp gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number pots`	Defines the POTS dial peer tag number and enter dial-peer configuration mode.
Step 2	`Router(config-dial-peer)# information-type fax`	Identifies calls associated with the dial peer as fax transmissions.
Step 3	`Router(config-dial-peer)# destination-pattern [+]string`	Identifies the destination fax telephone number.
Step 4	`Router(config-dial-peer)# prefix string`	(Optional) Specifies the prefix of the dialed digits associated with the dial peer. If a prefix is configured, the argument *string* is sent to the modem first, before the configured telephone number.

Configuring the MMoIP Dial Peer

To configure the off-ramp gateway MMoIP dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice number mmoip`	Defines the MMoIP dial peer tag number and enters dial-peer configuration mode
Step 2	`Router(config-dial-peer)# information-type fax`	Identifies calls associated with this dial peer as being fax transmissions, as opposed to strictly being voice calls.
Step 3	`Router(config-dial-peer)# incoming called-number string`	Identifies the destination fax telephone number.
Step 4	`Router(config-dial-peer)# image resolution {fine | standard | super-fine | passthrough}`	Specifies the fax image resolution for TIFF files associated with this particular MMoIP dial peer. **Note** Only standard and fine fax resolutions are supported.
Step 5	`Router(config-dial-peer)# image encoding {mh | mr | mmr | passthrough}`	Specifies the type of encoding to be used for TIFF files associated with this MMoIP dial peer.
Step 6	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode.

Note When configuring the MMoIP dial peer, ensure that the **incoming called number** command value and the configured destination telephone number (corresponding on-ramp POTS dial peer) match.

Configuring the Faxed Header Information

Because the off-ramp gateway does not alter fax TIFF attachments, the header information cannot be configured for faxes being converted from TIFF files to standard fax transmissions.

To configure faxed header information, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# fax send center-header {$a$ | $d$ | $p$ | $s$ | $t$ | string}`	Specifies the header information to be displayed in the center position. The keywords and arguments are as follows: • **d**—Specifies the destination address. • **s**—Specifies the sender address. • **p**—Specifies the page count. • **t**—Specifies the transmission time. • *string*—Inserts a personalized text string.

	Command	Purpose
Step 2	`Router(config)# fax send right-header {$a$ | $d$ | $p$ | $s$ | $t$ | string}`	Specifies the header information to be displayed on the right. Use the *string* argument in this command to insert a personalized text string.
Step 3	`Router(config)# fax send left-header {$a$ | $d$ | $p$ | $s$ | $t$ | string}`	Specifies the header information to be displayed on the left. Use the *string* variable in this command to insert a personalized text string.

Configuring the Fax Cover Page Information

Because the off-ramp gateway does not alter fax TIFF attachments, the cover pages cannot be configured for faxes being converted from TIFF files to standard fax transmissions.

To configure fax cover page information, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# fax send coverpage enable`	Enables the off-ramp gateway to send a cover sheet with faxes that originate from e-mail messages.
Step 2	`Router(config)# fax send coverpage comment string`	(Optional) Adds personalized text in the title field of the fax cover sheet.
Step 3	`Router(config)# fax send coverpage show-detail`	(Optional) Prints all of e-mail header information as part of the fax cover sheet text.
Step 4	`Router(config)# fax send coverpage enable`	(Optional) Enables the off-ramp gateway to send a cover page with faxes that originate from e-mail messages.
Step 5	`Router(config)# fax send coverpage e-mail controllable`	(Optional) Configures the router to defer to the cover page setting in the e-mail header. For example, if the address has a parameter set to cover=no or cover=yes, it will override the setting for the **fax send coverpage enable** command.

Verifying the Gateway Configuration

To verify the gateway configuration, perform the following tasks:

- Use **debug fax send calling-number** to check the transmitting subscriber number configuration.
- Use **debug fax send all** to display Class 2 fax protocol tracing information for all off-ramp faxing activities.
- Use **debug mta receive all** to view output relating to the activity on the SMTP server (messages exchanged; for example, the handshake) between the e-mail server and the off-ramp gateway.
- Use **debug text-to-fax** to view information relating to the off-ramp text-to-fax conversion.
- Use **debug tiff reader** to display output about the on-ramp TIFF reader.
- Use **debug tiff writer** to display output about the on-ramp TIFF writer.
- Send an e-mail message to the off-ramp gateway to check whether the fax cover page generates correctly.
- Send a fax-mail using a mail client to the off-ramp gateway and request a return receipt in the e-mail message to check if the fax-mail is processed correctly. The destination e-mail address must have the appropriate fax=user@receive alias to be allowed.

Configuring Gateway Security

To configure gateway security, perform the tasks in the following sections:

- Configuring On-Ramp Gateway Security, page 738 (Required)
- Configuring Off-Ramp Gateway Security, page 739 (Required)
- Configuring the Gateway Security for TCL Application Files, page 740 (Required)

Configuring On-Ramp Gateway Security

To configure on-ramp security, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new model`	Enables AAA security services.
Step 2	`Router(config)# mmoip aaa method fax authentication` *method-list-name*	Defines the name of the method list to be used for Store and Forward Fax AAA authentication.
Step 3	`Router(config)# mmoip aaa method fax accounting` *method-list-name*	Defines the name of the method list to be used for Store and Forward Fax AAA accounting.
Step 4	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Creates a local authentication method list and enables authentication.
Step 5	`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {stop-only} [method1 [method2...]]`	Creates an accounting method list and enables accounting. We recommend the following configuration: **aaa accounting connection** *list-name* **stop-only**.
Step 6	`Router(config)# mmoip aaa receive-id primary {ani | dnis | gateway | redialer-id | redialer-dnis}`	Specifies the primary location where AAA retrieves its identifying information for on-ramp faxing.
Step 7	`Router(config)# mmoip aaa receive-id secondary {ani | dnis | gateway | redialer-id | redialer-dnis}`	(Optional) Specifies the secondary location where AAA retrieves its identifying information for on-ramp faxing.
Step 8	`Router(config)# mmoip aaa receive-authentication enable`	Enables on-ramp AAA authentication services.
Step 9	`Router(config)# mmoip aaa receive-accounting enable`	Enables on-ramp AAA accounting services.
Step 10	`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number]`	Specifies the IP address or host name of the remote RADIUS server host and assigns authentication and accounting destination port numbers. Typical authentication and accounting destination ports are 1645 and 1646.
Step 11	`Router(config)# radius-server key string`	Specifies the shared-secret text string used between the router and the RADIUS server.
Step 12	`Router(config)# radius-server vsa send accounting`	Enables the network access server to recognize and use accounting VSAs as defined by RADIUS IETF attribute 26.
Step 13	`Router(config)# radius-server vsa send authentication`	Enables the network access server to recognize and use authentication VSAs as defined by RADIUS IETF attribute 26.

Configuring Off-Ramp Gateway Security

Note It is recommended that the off-ramp gateway (the packet filters) be configured to accept only incoming SMTP connections (IP addresses) from trusted mailers when faxes are sent to the off-ramp gateway.

To configure off-ramp security, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new model`	Enables AAA security services.
Step 2	`Router(config)# mmoip aaa method fax authentication` *method-list-name*	Defines the name of the method list to be used for Store and Forward Fax AAA authentication.
Step 3	`Router(config)# mmoip aaa method fax accounting` *method-list-name*	Defines the name of the method list to be used for Store and Forward Fax AAA accounting.
Step 4	`Router(config)# aaa authentication login {default | list-name} method1 [method2...]`	Creates a local authentication method list and enables authentication.
Step 5	`Router(config)# aaa accounting {system | network | exec | connection | commands level} {default | list-name} {stop-only} [method1 [method2...]]`	Creates an accounting method list and enables accounting. It is recommended that **aaa accounting connection** *list-name* **stop-only** be used.
Step 6	`Router(config)# mmoip aaa send-id primary {account-id | envelope-from | envelope-to | gateway}`	Specifies the primary location where AAA retrieves its identifying information for off-ramp faxing.
Step 7	`Router(config)# mmoip aaa send-id secondary {account-id | envelope-from | envelope-to | gateway}`	(Optional) Specifies the secondary location where AAA retrieves its identifying information for off-ramp faxing.
Step 8	`Router(config)# mmoip aaa send-authentication enable`	Enables off-ramp AAA authentication services.
Step 9	`Router(config)# mmoip aaa send-accounting enable`	Enables off-ramp AAA accounting services.
Step 10	`Router(config)# radius-server host {hostname | ip-address} [auth-port port-number] [acct-port port-number]`	Specifies the IP address or host name of the remote RADIUS server host and assigns authentication and accounting destination port numbers. Typical authentication and accounting destination ports are 1645 and 1646.
Step 11	`Router(config)# radius-server key string`	Specifies the shared secret text string used between the router and the RADIUS server.
Step 12	`Router(config)# radius-server vsa send accounting`	Enables the network access server to recognize and use accounting VSAs as defined by RADIUS IETF attribute 26.
Step 13	`Router(config)# radius-server vsa send authentication`	Enables the network access server to recognize and use authentication VSAs as defined by RADIUS IETF attribute 26.

Configuring the Gateway Security for TCL Application Files

To configure gateway security for the TCL application files that are used for fax calls on the T.37/T.38 Fax Gateway with a VFC, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call application voice application-name accounting enable`	Enables AAA accounting services for the named application.
Step 2	`Router(config)# call application voice application-name accounting-list method-list-name`	Defines the name of the method list to be used for AAA accounting with fax applications on a VFC.
Step 3	`Router(config)# call application voice application-name authentication enable`	Enables AAA authentication services for the named application.
Step 4	`Router(config)# call application voice application-name authen-list method-list-name`	Specifies the name of an authentication method list for the named application.
Step 5	`Router(config)# call application voice application-name authen-method id`	Specifies the name of the authentication method for the named application. Valid authentication ids are prompt-user, gateway, ani, dnis, redialer-id, and redialer DNIS.

Verifying the Gateway Security Configuration

To verify the gateway security configuration, perform the following tasks:

- Use the **debug mmoip aaa** command to verify that the on-ramp security is configured correctly.
- Check the console log file, depending upon the RADIUS version used, to verify connection to the RADIUS server.
- Use the **debug aaa** command to verify AAA performance.

Configuring MDNs

Note The MDN elements must be configured for both the on-ramp and off-ramp gateways.

To configure the on-ramp gateway to support MDN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mta send return-receipt-to username string`	Specifies the username address. If this field is left blank, the on-ramp gateway inserts the postmaster address in this field as a default.
Step 2	`Router(config)# mta send return-receipt-to hostname string`	Specifies the host name address. If this field is left blank, the on-ramp gateway inserts the postmaster address in this field as a default.

	Command	Purpose
Step 3	`Router(config)# dial-peer voice number mmoip`	Defines the MMoIP dial peer tag number and enters dial-peer configuration mode.
Step 4	`Router(config-dial-peer)# mdn`	Sends the MDN to the destination defined by the **mta send return-receipt-to** command.

To configure the off-ramp gateway to support MDN, use the following command in global configuration mode:

Command	Purpose
`Router(config)# mta receive generate-mdn`	Specifies that the Cisco AS5300 universal access server acting as the off-ramp gateway will respond to a request for an MDN.

Verifying MDN Configuration

To verify the MDN configuration, perform the following tasks:

- Verify if DSN is enabled or disabled using the **show dial-peer voice** command and look at the disposition notification field.
- Verify that the **mta send return-receipt-to username**, **mta send return-receipt-to hostname**, and **mta receive generate-mdn** commands have been configured by using the **show running-config** command.
- Send a fax to the on-ramp gateway. When the destination e-mail account client opens and responds to the MDN request, check the return-receipt-to user account for the MDN response message.
- Send a fax to the off-ramp gateway with MDN requested (return receipt). After the off-ramp gateway has processed the fax-mail message, check the original From: user's account for the MDN response message.

Configuring DSNs

Note The DSN elements must be configured for both the on-ramp and off-ramp gateways.

To configure DSN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# mta send mail-from {hostname string}`	Specifies the originator (host name portion) of the e-mail fax message. Use this command with the **mta send mail-from username** command to form a complete e-mail address (faxuser@onramp-gateway.com).

	Command	Purpose
Step 2	`Router(config)# mta send mail-from {username string | username $s$}`	Specifies the originator (username portion) of the e-mail fax message. The keyword **s** generates a transmission report that is sent to the originating fax machine. Use this command with the **mta send mail-from hostname** command to form a complete e-mail address (faxuser@onramp-gateway.com).
Step 3	`Router(config)# dial-peer voice number mmoip`	Defines the MMoIP dial peer tag number and enters dial-peer configuration mode.
Step 4	`Router(config-dial-peer)# dsn {delay | success | failure}`	Sends a DSN to the destination defined by the **mta send mail-from** command.

Verifying DSN Configuration

To verify the DSN configuration, perform the following tasks:

- Use the **show dial-peer voice** command and look at the delivery status notification field.
- Use the **show running-config** command to display the **mta send mail-from username** and **mta send mail-from hostname** configurations. If these commands are not configured, the DSN will be delivered to the postmaster defined by the **mta send postmaster** command.
- Use the **show running-config** command to display the **mta send return-receipt-to username**, **mta send return-receipt-to hostname**, and **mta receive generate-mdn** configurations.
- Send a fax to the on-ramp gateway. When the destination e-mail server receives the fax-mail message and responds to the DSN request, check the mail-from or postmaster user account for the DSN response message. (The mail-from or postmaster user account could be a fax machine.)
- Send a fax-mail message to the off-ramp gateway with DSN requested (rcpt to:<fax=555-1212@company.com> NOTIFY=SUCCESS, FAILURE, DELAY). After the off-ramp gateway has processed the fax-mail message, check the original From: user's account for the DSN response message.

Configuring T.37 Store and Forward Fax

The Cisco AS5300 universal access server supports only two modem cards: the Microcom modem card and the MICA technologies modem card. Microcom modem cards support both on-ramp and off-ramp fax activities. MICA technologies modem cards support only off-ramp faxing.

Store and forward fax on-ramp has been designed to work by using direct inward dial (DID) or a redialer. A redialer is a hardware interface device that interconnects between a fax device and the PSTN. If DID is disabled, a redialer must be configured and enabled on the originating fax machine before Store and Forward Fax is operational.

To configure the T.37 Store and Forward Fax application, configure the on- and off-ramp gateways, including gateway security, and perform the following tasks:

- Configuring On-Ramp Modem Pooling, page 743 (Required)
- Configuring ECM, page 743 (Required)

Configuring On-Ramp Modem Pooling

To configure on-ramp modem pooling, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# modem-pool name`	Creates a modem pool.
Step 2	`Router(config)# pool-range number-number`	Assigns a range of modems to the specified modem pool.

Configuring ECM

To configure ECM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice 99 voip`	Enters dial peer configuration mode for the VoIP dial peer.
Step 2	`Router(config-dial-peer)# fax-relay ecm disable`	Disables ECM. **Note** Use the **no fax-relay ecm disable** command to enable ECM.

Configuring the T.37/T.38 Fax Gateway

The Cisco AS5300 universal access server must be equipped with 128 MB of RAM in the following situations:

- When a maximum of 120 simultaneous Store and Forward Fax sessions is required
- If IVR Version 2.0 is required

To configure the T.37/T.38 Fax Gateway feature, configure the on- and off-ramp gateways, including gateway security and perform the following tasks:

- Specifying the Interface Type for Fax Calls, page 744 (Required)
- Configuring IVR Functionality, page 744 (Required)
- Configuring T.38 Fax Relay for VoIP H.323, page 746 (Required)

Specifying the Interface Type for Fax Calls

To select the interface type (modem or VFC), use the following command in global configuration mode:

Command	Purpose
`Router(config)# fax interface-type {modem | vfc}`	Specifies the interface type. **Note** On Cisco AS5300 access servers, the keyword **vfc** maps to the **fax-mail** keyword. If you enter the **show run** command, the **fax-mail** keyword will display. The defaults are determined as follows: •If the gateway has modem cards only, the default is the **modem** keyword. •If the gateway has voice cards only, the default is the **vfc** (**fax-mail)** keyword. The **modem** keyword is unavailable. This applies to all platforms except the Cisco AS5300 access server. •If the gateway has both modem and voice cards, the default is the **modem** keyword.

Configuring IVR Functionality

Note All IVR scripts are modified and secured with a proprietary Cisco locking mechanism. Only Cisco internal technical support personnel can open and modify these scripts. TCL must be installed before IVR functionality is configured.

To configure IVR functionality, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# call application voice` *application-name location*	Defines the name to be referenced and indicates the URL of the IVR script to be used. **Note** The *application-name* is a user-defined name which, once defined, is referenced in all other IVR commands except for **application** used with the on-ramp MMoIP dial peer.
Step 2	`Router(config)# call application voice` *application-name* **language** *language*	(Optional) Defines the language of the audio file and passes that information to the application.
Step 3	`Router(config)# call application voice` *application-name* **pin-length** *number*	(Optional) Defines the number of PIN characters and passes that information to the application.
Step 4	`Router(config)# call application voice` *application-name* **retry-count** *number*	(Optional) Defines the number of times a caller is permitted to reenter the PIN and passes that information to the application.
Step 5	`Router(config)# call application voice` *application-name* **uid-length** *number*	(Optional) Defines the number of UID characters and passes that information to the application.

	Command	Purpose
Step 6	`Router(config)# call application voice application-name set-location language category location`	(Optional) Defines the location, language, and category of the audio files and passes that information to the application.
Step 7	`Router(config)# aaa new-model`	Enables AAA security and accounting services.
Step 8	`Router(config)# gw-accounting h323`	Enables gateway-specific H.323 accounting.
Step 9	`Router(config)# aaa authentication login h323 radius`	Defines a method list called **h323** where in RADIUS is defined as the only method of login authentication.
Step 10	`Router(config)# aaa accounting connection h323 start-stop radius`	Defines a method list called **h323** where in RADIUS is used to perform connection accounting, providing start and stop records.
Step 11	`Router(config)# radius-server host ip-address auth-port number acct-port number`	Identifies the RADIUS server and the ports that will be used for authentication and accounting services.
Step 12	`Router(config)# radius-server key key`	Specifies the password used between the gateway and the RADIUS server.
Step 13	`Router(config)# dial peer voice number pots`	Changes mode to dial peer configuration.
Step 14	`Router(config-dial-peer)# port port number`	Defines the voice port associated with the POTS dial peer.
Step 15	`Router(config-dial-peer)# ctrl + z`	Exits to privileged EXEC mode.

Table 56 lists the TCL scripts required for fax applications on VFCs.

Table 56 TCL Scripts Required for VFCs

TCL Script Name	Description—Summary	Commands to Configure
app_libretto_onramp9.tcl	Authenticates the account and PIN using the following: prompt-user, ANI, DNIS, gateway ID, redialer ID, and redialer DNIS.	None
app_libretto_offramp5.tcl	Authenticates the account and PIN using the following: envelope-from, envelope-to, gateway ID, and x-account ID.	None
fax_rollover_on_busy.tcl	Used for on-ramp T.38 fax rollover to T.37 fax when the destination fax line is busy.	**voice hunt user-busy**

Verify the IVR Configuration

To verify the IVR configuration, perform the following tasks:

- Use the **show running-config** to verify the configuration parameters.
- Use the **show call application summary** to display a list of all voice applications.
- Use the **show call application voice** to display the contents of the script.
- Use the **show dial-peer voice** to verify that a dial peer is operational.

Configuring T.38 Fax Relay for VoIP H.323

Only User Datagram Protocol (UDP) is implemented for T.38 Fax Relay for VoIP H.323 gateway support on the multiservice gateways. Transmission Control Protocol (TCP) T.38 Fax Relay is not supported. For further information on T.38 protocol, refer to ITU-T Recommendation.

Voice interoperability testing with third-party gateways and gatekeepers must be completed before configuring the T.38 Fax Relay for VoIP H.323 in the network because different companies are allowed to select certain parts of H.323 and T.38 to implement into their gateways and gatekeepers.

T.38 Fax Relay interoperability requires H.323 Version 2. In addition, T.38 Fax Relay is not supported in the following:

- Cisco MC3810 multiservice concentrators with VCM (Voice Compression Module)
- T.38 Fax Relay is not supported by Multimedia Conference Manager (MCM) H.323 proxy
- T.38 Fax Relay is not supported in conjunction with Media Gateway Control Protocol (MGCP), Simple Gateway Control Protocol (SGCP), or Session Initiation Protocol (SIP)

Configure both the on-ramp and off-ramp gateways to enable T.38 Fax Relay for VoIP H.323. To specify the global default fax protocol for all the VoIP dial peers, use the global configuration mode. To specify the fax protocol for a specific VoIP dial peer, which takes precedence over the global configuration, use dial-peer configuration mode.

Configuring T.38 Fax Relay for VoIP H.323 Globally

To configure T.38 Fax Relay for VoIP H.323 for all the connections of a gateway, which is required, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice service voip`	Enters voice-service configuration mode.
Step 2	`Router(config-voi-serv)# fax protocol {cisco | t38 [ls_redundancy value] [hs_redundancy value]}`	Specifies the global default fax protocol. The keywords and arguments are as follows: • **cisco**—Selects the original Cisco proprietary fax protocol. • **t38**—Enables the T.38 Fax Relay protocol. • **ls_redundancy**—(Optional) Sends redundant T.38 fax packets in the low-speed V.21-based T.30 fax machine protocol. • *value*—Specifies redundancy from 0 to 5. The default is 0 (no redundancy). • **hs_redundancy**—Sends redundant T.38 fax packets in the high-speed V.17, V.27, and V.29 T.4 or T.6 fax machine image data. • *value*—Specifies redundancy from 0 to 2. The default is 0 (no redundancy). If set to a value greater than zero, network bandwidth could be increased or consumed.

Configuring T.38 Fax Relay for a Specific Dial Peer

To configure T.38 Fax Relay for VoIP H.323 for a specific dial peer, which is optional, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice tag voip`	Enters dial-peer configuration mode.
Step 2	`Router(config-dial-peer)# fax protocol {cisco | t38 [ls_redundancy value] [hs_redundancy value] | system}`	Specifies the fax protocol for a dial peer. The keywords and arguments are as follows: • **cisco**—Selects the original Cisco proprietary fax protocol. • **t38**—Enables the T.38 Fax Relay protocol. • **ls_redundancy**—(Optional)Sends redundant T.38 fax packets in the low-speed V.21-based T.30 fax machine protocol. • *value*—Specifies redundancy from 0 to 5. The default is 0 (no redundancy).

	Command	Purpose
		• **hs_redundancy**—Sends redundant T.38 fax packets in the high-speed V.17, V.27, and V.29 T.4 or T.6 fax machine image data. • *value*—Specifies redundancy from 0 to 2. The default is 0 (no redundancy). If set to a value greater than zero, network bandwidth could be increased or consumed. • **system**—Specifies the global default fax protocol used by a dial peer, set by the **fax protocol t.38** command.
Step 3	`Router(config-dial-peer)# fax rate {12000 | 14400 | 2400 | 4800 | 7200 | 9600} {disable | voice} [bytes bytes]`	Selects the maximum fax transmission speed.

Verifying T.38 Fax Relay for VoIP H.323

To verify the T.38 fax relay for VoIP H.323, perform the following tasks:

- Enter the **show running-config** command.
- Enter the **show dial-peer voice** command.

Troubleshooting Tips

To troubleshoot the T.38 Fax Relay for VoIP H.323 feature, perform the following steps:

- Ensure that a voice call can be made.
- Ensure that T.38 Fax Relay for VoIP H.323 is configured on both the on-ramp and off-ramp gateways.
- Ensure that the fax protocol is configured as T.38 in global configuration mode or dial-peer configuration mode for both the on-ramp and off-ramp gateways.
- Use the **debug vtsp session**, **debug cch323 session**, and the **debug cch323 h245** commands to debug a problem.
- Use the **debug voip ccapi inout** command to debug problems while making the call.

Monitoring and Maintaining T.38 Fax Relay for VoIP H.323

To monitor and maintain the T.38 fax relay for VoIP H.323, perform the following tasks:

- Use the **show running-config** command to display the current configuration.
- Use the **show dial-peer voice summary** command to display the configuration information for all dial peers.

Fax Applications Configuration Examples

Configuration examples are provided in the following sections:

T.37 Store and Forward Fax Configuration Examples

The following output sample is the configuration of a Cisco AS5300 universal access server acting as an on-ramp gateway in global configuration mode:

```
!Define the called subscriber number. In this case, the number configured as the
!destination pattern will be used as the called subscriber identifier.
 fax receive called-subscriber $d$
!
!Specify the originator of the e-mail address. In this case, the originator information
!is derived from the calling number.
 mta send mail-from username $s$
!
!(Optional) Provide additional information about the sending device. In this example,
!the sending device's hostname is alabama
 mta send origin-prefix alabama
!
!Define where this fax-mail should be delivered (which is the mail server postmaster
!account) if it cannot be delivered to the defined destination.
 mta send postmaster postmaster@company.com
!
!(Optional) If configuring MDNs, specify the address where they should be
!sent.
 mta send return-receipt-to username postmaster@company.com
!
!Specify the destination e-mail server that accepts on-ramp fax-mail.
 mta send server california.fax.com
!
!Define the text string that will be displayed as the subject of the fax-mail.
 mta send subject Fax-Mail Message
!
!
!Enter dial-peer configuration mode and define an on-ramp POTS peer.
dial-peer voice 1000 pots
!
!Designate fax as the type of information handled by this dial peer.
 information-type fax
!
!Specify direct inward dial for this dial peer.
 direct-inward-dial
!
!Define the incoming called number associated with this dial peer
 incoming called number 5105551212
!
!(Optional) Define the maximum number of connections that will be used simultaneously
!to transmit fax-mail.
 max-conns 10
!
!
!Define an on-ramp MMoIP dial peer.
dial-peer voice 1001 mmoip
!
!Define the telephone number associated with this dial peer.
```

```
 destination-pattern 14085554321
!
!Define a destination e-mail address for this dial peer
 session-target mailto:$d$@abccompany.com
!
!(Optional) Request that DSNs be sent.
 dsn
!
!Specify a particular image encoding method to be used for fax images. In this
!example, Modified Huffman (IETF standard) is being specified.
 image encoding mh
!
!Specify a particular fax image resolution. In this example, the image resolution was
!set to 204 by 196 pixels per inch (fine).
 image resolution fine
!
!Designate fax as the type of information handled by this dial peer.
 info-type fax
!
!(Optional) Define the maximum number of connections that will be used simultaneously
!to transmit fax-mail.
 max-conn 10
!
!(Optional) Request that MDNs be sent.
 mdn
!
!Specify SMTP as the protocol to be used for Store and Forward Fax.
 session protocol smtp
```

The following output sample is the configuration of a Cisco AS5300 universal access server acting as the off-ramp gateway beginning in global configuration mode:

```
!Define the transmitting subscriber number (TSI); this is the number that is
!displayed in the LCD of the receiving fax machine. In this example, the sender's
!name, captured by the on-ramp from the sending fax machine) will be used.
 fax send transmitting-subscriber $s$
!
!Configure the speed of the fax transmission. In this case, fax transmissions will be
!sent at 14400 bits per second.
 fax send max-speed 14400
!
!Define a hostname to be used as an alias for the off-ramp Cisco AS5300 device.
 mta receive aliases abccompany.com
!
!(Optional) Specify that the Cisco AS5300 universal access server will respond to an MDN
request.
 mta receive generate-mdn
!
!Define the number of simultaneous SMTP recipients (in this case, 10) handled by this
!Cisco AS5300 device.
 mta receive maximum-recipients 10
!
!
!Specify that the company name will appear in the center position of the fax.
!header information.
 fax send center-header Acme Company
!
!Specify that the page count will appear in the right position of the fax header
!information.
 fax send right-header $p$
!
!Specify that the date will appear in the left position of the fax header
!information.
 fax send left-header $a$
```

```
!
!Enable the Cisco AS5300 device to send a cover sheet with faxes that originate from
!e-mail messages.
 fax send coverpage enable
!
!Add a personalized comment to the title field of the fax cover sheet. In this case,
!the phrase FAX TRANSMISSION was added.
fax send coverpage comment FAX TRANSMISSION
!
!Enter dial-peer configuration mode and define an off-ramp POTS peer.
dial-peer voice 1002 pots
!
!Designate fax as the type of information handled by this dial peer.
 information-type fax
!
!Define a telephone number to be associated with this dial peer.
 destination-pattern 1408555....
!
!Add prefix.
prefix 9,555
!
!Define an off-ramp MMoIP peer.
dial-peer voice 1003 mmoip
!
!Designate fax as the type of information handled by this dial peer.
 information-type fax
!
!Define an incoming called number to be associated with this dial peer.
 incoming called-number 14085556789
!
!Specify a particular fax image resolution. In this example, the image resolution was
!set to 204 by 196 pixels per inch (fine).
 image resolution fine
```

The following sample output is the configuration of the on-ramp and off-ramp gateway for security in global configuration mode:

```
!Enable AAA security services.
 aaa new-model
!Define the method list to be used with Store and Forward Fax authentication.
 mmoip aaa method fax authentication onramp-auth
!Define the method list to be used with Store and Forward Fax accounting services.
 mmoip aaa method fax accounting onramp-acct
!Define and enable the AAA authentication method list for Store and Forward Fax.
 aaa authentication login onramp-auth radius local
!Define and enable the AAA accounting method list for Store and Forward Fax.
 aaa accounting connection onramp-acct stop-only radius
!Enable on-ramp authentication.
 mmoip aaa receive-authentication enable
!Enable on-ramp accounting services.
 mmoip aaa receive-accounting enable
!Enable off-ramp authorization.
 mmoip aaa send-authentication enable.
!Enable off-ramp accounting services.
 mmoip aaa receive-accounting enable
!Define the gateway ID as the means by which AAA identifies the user for
!off-ramp authentication.
mmoip aaa send-id primary gateway
!Define the gateway ID as the means by which AAA identifies the user for on-ramp
!authentication.
mmoip aaa receive-id primary gateway
!Configure the Cisco AS5300 device to support RADIUS.
 radius-server host 173.13.11.13 auth-port 1645 acct-port 1646
 radius-server key password
```

```
!Configure the RADIUS server to recognize and use vendor-specific attributes.
 radius-server vsa send accounting
 radius-server vsa send authentication
```

The following sample output is the configuration of the on-ramp modem pool that uses 24 Microcom and 60 MICA modems in global configuration mode:

Note Microcom modems are in slot 1 and MICA modems are in slot 2. The purpose of this named modem pool (mica-inbound) is to prevent fax calls from going to the MICA modems (modems 25 through 84).

```
modem-pool mica-inbound
pool-range 25-84
```

The following sample output is complete Cisco AS5300 universal access server configuration:

```
Router# show running-config

Building configuration...

Current configuration:
!
!Last configuration change at 19:20:39 PST Mon Jul 14 1997
!NVRAM config last updated at 19:11:04 PST Mon Jul 14 1997
!
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
service tcp-small-servers
!
hostname mmoip-b
!
boot system tftp /auto/annex2/njoffe/c5300-is-mz 255.255.255.255
boot system flash c5300-is-mz
aaa new-model
aaa authentication login fax radius local
aaa accounting connection fax stop-only radius
!
username njoffe password 0 password
username jfitzhug password 0 password
username wooksong password 0 password
username gmercuri password 0 password
username faryaman password 0 password
username ilyau password 0 password
clock timezone PST -8
clock calendar-valid
!
modem-pool mica-inbound
modem poll time 2
ip subnet-zero
ip host mail-server 10.14.116.1
ip host keyer 223.255.254.254
ip host mail-server.cisco.com 10.14.116.1
ip domain-name cisco.com
ip name-server 10.14.116.1
!
```

```
isdn switch-type primary-5ess
fax receive called-subscriber $d$
fax send transmitting-subscriber $s$
fax send left-header $s$
fax send center-header $t$
fax send right-header Page:$p$
fax send coverpage enable
fax send coverpage email-controllable
fax send coverpage comment Cisco cover page comment
mta send server mail-server.cisco.com
mta send subject mmoip-b subject line here
mta send origin-prefix Cisco Powered Fax System
mta send postmaster gmercuri@mail-server.com
mta send mail-from hostname mail-from-hostname.com
mta send mail-from username $s$
mta send return-receipt-to hostname mmoip-b.cisco.com
mta send return-receipt-to username $s$
mta receive aliases mmoip-b.cisco.com
mta receive aliases [1.2.3.4]
mta receive aliases cisco.com
mta receive maximum-recipients 24
mta receive generate-mdn
mmoip aaa send-id primary gateway
mmoip aaa receive-id primary gateway
mmoip aaa method fax authentication fax
mmoip aaa method fax accounting fax
mmoip aaa send-accounting enable
mmoip aaa send-authentication enable
mmoip aaa receive-accounting enable
mmoip aaa receive-authentication enable
!
controller T1 0
framing esf
clock source line primary
linecode b8zs
cablelength short 133
pri-group timeslots 1-24
!
controller T1 1
shutdown
framing esf
clock source line secondary 1
linecode b8zs
cablelength short 133
cas-group 0 timeslots 1-24 type e&m-fgb service fax
!
controller T1 2
shutdown
framing esf
linecode b8zs
cablelength short 133
cas-group 0 timeslots 1-24 type e&m-fgb
!
controller T1 3
shutdown
framing esf
linecode b8zs
cablelength short 133
pri-group timeslots 1-24
!
voice-port 0:D
timeouts call-disconnect 0
!
```

```
voice-port 1:0
timeouts call-disconnect 0
!
voice-port 2:0
timeouts call-disconnect 0
!
voice-port 3:D
timeouts call-disconnect 0
!
dial-peer voice 5 mmoip
destination-pattern 55508..
information-type fax
mdn
dsn success
dsn failure
session target mailto:$d$@mail-server.cisco.com
!
dial-peer voice 1001 pots
incoming called-number 571....
port 0:D
!
dial-peer voice 2 pots
incoming called-number 5550839
information-type fax
direct-inward-dial
!
dial-peer voice 1 pots
destination-pattern 5......
information-type fax
prefix 5
!
num-exp 01133...... 33.......
!
interface Loopback0
no ip address
no ip directed-broadcast
!
interface Tunnel1
no ip address
no ip directed-broadcast
!
interface Ethernet0
ip address 10.14.120.2 255.255.0.0
no ip directed-broadcast
!
interface Serial0:23
no ip address
no ip directed-broadcast
encapsulation ppp
no ip route-cache
dialer rotary-group 1
dialer-group 1
isdn switch-type primary-5ess
isdn tei-negotiation first-call
isdn incoming-voice modem
no fair-queue
!
interface Serial3:23
no ip address
no ip directed-broadcast
shutdown
isdn switch-type primary-5ess
isdn tei-negotiation first-call
no cdp enable
```

```
!
interface FastEthernet0
no ip address
no ip directed-broadcast
shutdown
!
interface Group-Async1
ip unnumbered Ethernet0
no ip directed-broadcast
encapsulation ppp
ip tcp header-compression
dialer in-band
dialer-group 1
async mode interactive
peer default ip address pool default
no fair-queue
ppp multilink
group-range 1 12
hold-queue 10 in
!
interface Dialer1
ip unnumbered Loopback0
no ip directed-broadcast
encapsulation ppp
dialer in-band
dialer-group 1
peer default ip address pool def
no fair-queue
ppp multilink
!
ip default-gateway 10.14.0.1
no ip http server
ip classless
ip route 223.255.254.0 255.255.255.0 10.14.0.1
!
dialer-list 1 protocol ip permit
snmp-server engineID local 00000009020000E01EA48784
snmp-server community public RW
radius-server host 10.14.116.1 auth-port 1645 acct-port 1646
radius-server key password
radius-server vsa send accounting
radius-server vsa send authentication
!
line con 0
exec-timeout 0 0
logging synchronous
transport input none
line 1 12
autobaud
autoselect ppp
modem InOut
modem autoconfigure type microcom_hdms
rotary 1
transport input all
line aux 0
line vty 0 4
exec-timeout 0 0
password password
!
exception core-file /auto/annex2/gmercuri/coredump
exception dump 223.255.254.254
ntp source Ethernet0
ntp update-calendar
ntp peer 223.255.254.254
```

```
scheduler heapcheck process
scheduler interval 1000
end
```

T.37/T.38 Fax Gateway Examples

The following output sample shows the configured VFCs on a Cisco AS5300 universal access server:

```
!version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
service udp-small-servers
service tcp-small-servers

hostname fax-gateway

aaa new-model
aaa authentication login fax group radius local
aaa authorization exec fax group radius
aaa accounting connection fax stop-only group radius
enable password lab

username betatest password 0 password

ip subnet-zero
ip host dirt 223.255.254.254
ip domain-name cisco.com
ip name-server 1.14.116.1

mgcp package-capability trunk-package
mgcp default-package trunk-package
isdn switch-type primary-5ess
isdn voice-call-failure 0
```

The following output sample shows the PSTN Fallback from the T.38 Gateway to the T.37 gateway after configuring the **voice hunt user-bus**y command. The global service is displayed first as in the following example:

```
voice service voip
 fax protocol t38 ls_redundancy 0 hs_redundancy 0

call application voice app_libretto_offramp5
tftp://dirt/libretto-test/app_libretto_offramp5.tcl
call application voice app_libretto_offramp5 authen-list fax
call application voice app_libretto_offramp5 authen-method gateway
call application voice app_libretto_offramp5 accounting-list fax

call application voice app_onramp9 tftp://dirt/libretto-test/app_libretto_onramp9.tcl
call application voice app_onramp9 authen-list fax
call application voice app_onramp9 authen-method gateway
call application voice app_onramp9 language 1 en
call application voice app_onramp9 accounting-list fax
call application voice app_onramp9 set-location en 0 tftp://dirt/cchiu/WV/en_new/

fax receive called-subscriber $d$
fax send transmitting-subscriber $s$
fax send left-header $s$
fax send center-header $t$
fax send right-header Page: $p$
fax send coverpage enable
```

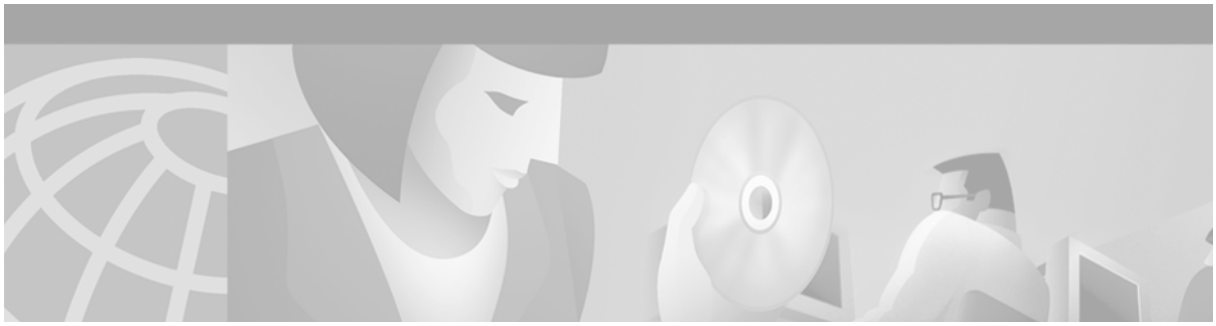
```
fax send coverpage email-controllable
fax send coverpage comment Cisco cover page comment
fax interface-type vfc
mta send server 1.14.116.1
mta send subject faxmail subject line here
mta send origin-prefix Cisco Powered Fax System
mta send postmaster postmaster@mail-server.cisco.com
mta send mail-from hostname fax-gateway.com
mta send mail-from username fax-user
mta send return-receipt-to hostname return.host.com
mta send return-receipt-to username $s$
mta receive aliases mmoip-b.cisco.com
mta receive aliases cisco.com
mta receive aliases [1.14.120.2]
mta receive maximum-recipients 80
mta receive generate-mdn

controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24

interface Ethernet0
 ip address 1.14.120.2 255.255.0.0
 no ip directed-broadcast

interface Serial0:23
 no ip address
 no ip directed-broadcast
 no ip route-cache
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no fair-queue

interface FastEthernet0
 no ip address
 no ip directed-broadcast
 shutdown
 duplex auto
 speed auto

ip default-gateway 1.14.0.1
ip classless
ip route 223.255.254.0 255.255.255.0 1.14.0.1
no ip http server

radius-server host 1.14.116.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key password
radius-server vsa send accounting
radius-server vsa send authentication

voice-port 0:D
 no modem passthrough

!Inbound Peer of the T.37 On-Ramp Gateway
dial-peer voice 2 pots
    application app_onramp9
     incoming called-number 5......
     direct-inward-dial
!Outbound Peer of the T.37 On-ramp Gateway
dial-peer voice 3 mmoip
!MDN and DSN Configuration of the Outbound Peer
```

```
 application fax_on_vfc_onramp_app out-bound
     destination-pattern 57108..
     session target mailto:$d$@mail-server.cisco.com
!Inbound Peer of the T.37 Off-Ramp Gateway
dial-peer voice 21 mmoip
     application lib_off_app5
     incoming called-number 5......
     information-type fax
!Outbound Peer of the T.37 Off-Ramp Gateway
!POTS 20 peer has port 0:D which means that when this peer is matched, controller T1-0 is
!used for the outgoing call:
dial-peer voice 20 pots
     destination-pattern 5......
     port 0:D
     prefix 5
!Inbound Peer for T.38 On-ramp Gateway
dial-peer voice 50 pots
     incoming called-number 1800555....
!Outbound Peer for On-Ramp Gateway
    dial-peer voice 51 voip
     destination-pattern 57108..
     session target ipv4:12.22.95.20
!Inbound Peer for Off-Ramp Gateway
    dial-peer voice 61 voip
     incoming called-number 57108..
!Outbound Peer for Off-Ramp Gateway
    dial-peer voice 60 pots
     destination-pattern 57108..
     port 0:D
     prefix 57108
!On-Ramp T.38 Fax Rollover to T.37
!Voice hunt user-busy is set first.
!Inbound peer of the T.37/T.38 on-ramp gateway
    dial-peer voice 70 pots
     application app_lib_rollover15
     incoming called-number 5......
!Outbound peer of the T.38 on-ramp gateway:
    dial-peer voice 71 voip
     preference 1
     destination-pattern 3746096
     session target ipv4:1.14.120.109
     fax protocol t38 ls_redundancy 0 hs_redundancy 0
!Outbound peer of the T.37 on-ramp gateway:
    dial-peer voice 72 mmoip
     preference 2
    application fax_on_vfc_onramp_app out-bound
     destination-pattern 3746096
     session target mailto:$d$@mail-server.cisco.com

    line con 0
     exec-timeout 0 0
     transport input all
    line aux 0
    line vty 0 4
     exec-timeout 0 0
     password password
    end
```

T.38 Fax Relay for VoIP H.323 Configuration Example

This section provides configuration examples of T.38 Fax Relay:

```
Router# show running-config
Building configuration...

Current configuration:
.
.
.
voice service voip
 fax protocol t38

.
.
.

interface Ethernet0/0
 ip address 10.0.47.47 255.255.0.0
 h323-gateway voip interface
 h323-gateway voip id ipaddr 10.0.47.36 1719
 h323-gateway voip h323-id 36402

.
.
.
dial-peer voice 14151 voip
!Uses t38 fax from voice service voip
 destination-pattern 14151..
 session target ras

dial-peer voice 14152 voip       !!! Uses Cisco fax for a specific dial peer
 destination-pattern 14152..
 session target ras
 fax protocol cisco

gateway
end
```



Configuring Video Applications

This chapter describes how to configure video support. It contains the following sections:

Note This chapter does not describe how to configure Multimedia Conference Manager. For more information, see the "Configuring H.323 Gatekeepers and Proxies" chapter.

For a complete description of the video application commands used in this chapter, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation for other commands that appear in this chapter, use the command reference master index or search online.

To identity the hardware platform or software image information mentioned in this appendix, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Video Applications Overview

This section contains the following subsections:

Cisco Video Support by Platform

Cisco video support for various applications is listed by platform in the following sections:

Cisco MC3810 Multiservice Concentrator

The Cisco MC3810 multiservice concentrator supports video traffic within a data stream in the following ways:

- Video in pass-through mode—By this method, video traffic received from a video codec connected to a universal I/O serial port can be transported on a dedicated time slot between systems using the time-division multiplexing (TDM) functionality of the T1/E1 trunk.
- Video over ATM adaptation layer 1 (AAL1)—A serial stream from a video codec connected to a Cisco MC3810 on serial port 0 or 1 can be converted to ATM and transported across an ATM network using AAL1 circuit emulation services (CES) encapsulation.
- Video over ATM permanent virtual circuits (PVCs) and switched virtual circuits (SVCs)—A serial stream from a video codec connected to a Cisco MC3810 using the plug-in video dialing module (VDM) can be converted to ATM and transported across an ATM network using AAL1 CES encapsulation.

Note Before configuring your MC3810 multiservice concentrator to support video traffic, you must first configure the clock source for the Cisco MC3810 interfaces. For more information, refer to the "Configuring Synchronized Clocking" appendix.

Note Only V.35 cable is supported for video traffic over serial port 0 or 1.

Cisco 2600 Series, 3600 Series, and 7200 Series Router and MC3810 Multiservice Concentrator

Cisco 2600 series, 3600 series, and 7200 series routers and the MC3810 multiservice concentrator support Multimedia Conference Manager with voice gateway image and Resource Reservation Protocol (RSVP) to ATM SVC mapping. Multimedia Conference Manager delivers H.323 gatekeeper, proxy, and voice gateway solutions with routing as a single Cisco IOS image. In addition, Multimedia Conference Manager enables H.323 RSVP reservations to be mapped to ATM nonreal-time variable bit rate (nRTVBR) SVCs to guarantee quality of service (QoS) for video applications over ATM backbones.

Cisco 3600 Series Router

Circuit emulation is a service based on ATM Forum standards that allows communications to occur between AAL1 CES and ATM user network interfaces (UNIs), that is, between non-ATM telephony devices (such as classic PBXs or time-division multiplexers) and ATM devices (such as Cisco 3600 series routers). Thus, a Cisco 3600 series router equipped with an OC-3/STM-1 ATM CES network module offers a migration path from classic T1/E1 data communications service to emulated CES T1/E1 unstructured (clear channel) services or structured (N x 64) services in an ATM network.

The OC-3/STM-1 ATM CES network module uses the CES clock and passes the clocking information to the T1 and E1 controller and to the ATM interface.

For specific information regarding OC-3/STM-1 ATM CES network module configurations, refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Multimedia Conference Manager with Voice Gateway Image and RSVP to ATM SVC Mapping

Multimedia Conference Manager with voice gateway image and RSVP to ATM SVC mapping is implemented on Cisco IOS software. Multimedia Conference Manager is supported on the Cisco 2600 series , 3600 series, and 7200 series routers and on the MC3810 multiservice concentrator.

Multimedia Conference Manager with voice gateway image and RSVP to ATM SVC mapping enables you to limit the H.323 traffic on the LAN and WAN; it provides user accounting for records based on the service use; it guarantees QoS for the H.323 traffic generated by applications such as Voice over IP (VoIP), data conferencing, and videoconferencing; and it guarantees the implementation of security for H.323 communications. In addition, this new and separate image also incorporates Cisco voice gateway and routing functionalities in the same image.

With voice gateway image and RSVP to ATM SVC mapping, you can stipulate bandwidth limits for each videoconferencing connection and an aggregate bandwidth limit for all videoconferencing sessions. This voice gateway image allows you to provide bandwidth limitation to the endpoints.

Additional benefits include the following:

- The proxy can forward T.120 connections, which enhances real-time data conferencing capabilities.
- The gatekeeper can perform load-balancing functionality for external H.323 Version 2 gateways.
- The gatekeeper supports call accounting for proxied calls. Proxied calls are recorded into call history to provide additional call detail information.

Multimedia Conference Manager is recommended for multiple Cisco CallManagers or CallManager cluster domains. Multimedia Conference Manager provides critical connection admission control (CAC) between domains to guarantee that the number of calls between locations does not exceed available bandwidth.

For more detailed information about Multimedia Conference Manager, see the “Configuring H.323 Gatekeepers and Proxies” chapter.

ATM Nonreal-Time VBR SVC Support for Video

ATM nonreal-time variable bit rate (nRTVBR) SVC service operates much like X.25 SVC service although ATM allows much higher throughput. Virtual circuits are created and released dynamically, providing user bandwidth on demand. This service requires a signaling protocol between the router and the switch. Each ATM node is required to establish a separate connection to every other node in the ATM network with which it needs to communicate. All such connections are established using a PVC or an SVC with an ATM signaling mechanism.

Using ATM nRTVBR SVC for video on an ATM backbone guarantees that video sessions will traverse that backbone with QoS features enabled. The Cisco IOS image takes H.323 RSVP reservations and maps them to ATM nRTVBR SVCs that are dynamically established and torn down when video sessions are established and terminated. End-to-end IP routing across the network backbone is no longer required to guarantee video QoS.

ATM nonreal-time nRTVBR SVC service is supported on the Cisco 2600 series, 3600 series, and 7200 series routers and on the MC3810 multiservice access server.

For more information on configuring ATM, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.

Video Applications Prerequisite Tasks and Restrictions

The following prerequisites and restrictions apply when using Multimedia Conference Manager with voice gateway image and RSVP to ATM SVC mapping:

- Permanent virtual pathways (PVPs) are supported only on OC-3 cards and DS3/E3 cards. Neither the T1-IMA cards nor the T1 interface on the Cisco MC3810 supports PVPs.
- T.120 proxy has been tested and proved to work with Microsoft NetMeeting 3.01. Based on testing, T.120 proxying does not work with VCON endpoints. T.120 proxy works only with endpoints that can connect to ports other than the default port of 1503. Microsoft NetMeeting 3.01 can do this, but VCON cannot.
- Some older H.323 endpoint implementations, especially those used in videoconferencing, may not be able to connect to an H.225 call setup port number other than 1720. If you have to use those endpoints with the H.323 gatekeeper proxy feature, consider using an image without the Cisco H.323 VoIP gateway (an -ix- image).
- If you are going to map RSVP requests to ATM nRTVBR SVCs, do not use the **ip flow-cache feature-accelerate** command. This command causes traffic from a reserved SVC to switch to a PVC when an ATM interface shuts down and comes back online.
- ATM-25 cards have not been tested for interoperability with this feature.
- For Multimedia Conference Manager with voice gateway image and RSVP to ATM SVC mapping to function properly, you must have 16 megabytes of Flash memory and 64 megabytes of DRAM memory. For the Cisco 3660 router and for the Cisco 7200 series router, 96 megabytes of DRAM are required.

Video Applications Configuration Task List

Video applications require different tasks. To configure video support, perform one of the following:

Configuring Video in Pass-Through Mode

Video in pass-through mode is supported on the Cisco MC3810 multiservice concentrator.

To configure support for video in pass-through mode, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# network-clock base-rate {56k | 64k}`	Configures the network clock base rate. The default is 56 kbps.
Step 2	`Router(config)# interface serial number {multipoint | point-to-point}`	Enters serial interface configuration mode for either serial port 0 or 1. • **multipoint**—Assumes that there is a fully meshed network. • **point-to-point**—Specifies that a video connection will be over a point-to-point network.
Step 3	`Router(config-if)# encapsulation clear-channel`	Configures the serial interface to be in clear-channel mode for pass-through traffic.
Step 4	`Router(config-if)# clock rate network-clock rate`	Configures the network clock speed for serial port 0 or 1 in DCE mode on the MC3810 multiservice access server. The *rate* argument is the network clock speed in bits per second. The range is from 56 kbps to 2048 kbps. The value entered should be a multiple of the value set for the **network-clock base-rate** command. The maximum rate supported is 2048 Mbps.

	Command	Purpose
Step 5	`Router(config-if)# exit`	Exits interface configuration mode.
Step 6	`Router(config)# controller t1 0`	Enters controller configuration mode for controller T1 0.
Step 7	`Router(config-controller)# tdm-group` *tdm-group-no* **timeslot** *timeslot-list* [**type** {**e&m** | **fxs** [**loop-start** | **ground-start**] | **fxo** [**loop-start** | **ground-start**] | **fxs-melcas** | **fxo-melcas** | **e&m-melcas**}]	Configures a list of time slots for creating clear channel groups (pass-through) for time-division multiplexing (TDM) cross-connect. The keywords and arguments are as follows: • *tdm-group-no*—Specifies the TDM group number. • **timeslot**—Specifies the timeslot number. • *timeslot-list*—Specifies the timeslot list. The valid range is from 1 to 24 for T1, and it is from 1 to 15 and 17 to 31 for E1. • **type**—(Optional) (Valid only when the **mode cas** command is enabled.) Specifies the voice signaling type of the voice port. If configuring a TDM group for data traffic only, do not specify the **type** keyword. Choose from one of the following options: – **e&m**—Specifies E&M signaling. – **fxs**—Specifies Foreign Exchange Office (FXO) signaling. (Optionally, you can also specify **loop-start** or **ground-start.**) – **fxo**—Specifies Foreign Exchange Station (FXS) signaling. (Optionally, you can also specify **loop-start** or **ground-start.)** – **fxs-melcas**—Specifies FXS Mercury Exchange Limited (MEL) Channel Associated Signaling (CAS). – **fxo-melcas**—Specifies FXO MEL CAS. • **e&m-melcas**—Specifies ear and mouth (E&M) MEL CAS.
Step 8	`Router(config-controller)# exit`	Exits controller configuration mode.

	Command	Purpose
Step 9	`Router(config)# cross-connect id controller-1 tdm-group-no-1 controller-2 tdm-group-no-2`	Cross-connects two groups of digital signal level 0s (DS0s) from two controllers on the Cisco MC3810 or cross-connects the Universal I/O (UIO) serial port for pass-through traffic to a trunk controller. Configures cross-connect pass-through from Universal I/O (UIO) serial port 0 or 1 to a controller. The arguments are as follows: • *id*—Specifies the unique identification (ID) assigned to this cross-connection. The valid range is from 0 to 31. • *controller-1*—Specifies the type of the first controller (T1 0, T1 1, or E1). • *tdm-group-no-1*—Specifies the time-division multiplexing (TDM) group number assigned to the first controller. • *controller-2*—Specifies the type of the second controller (T1, E1 0, or E1 1). • *tdm-group-no-2*—Specifies the TDM group number assigned to the second controller.

Configuring Video over ATM AAL1

This section describes how to configure video over ATM AAL1 PVCs using CES. This functionality does not use the VDM, and SVCs are not supported. This section describes the video functionality supported on the MC3810 multiservice concentrator.

To configure video support over ATM AAL1 PVCs on a Cisco 3600 series router, see the "Configuring Structured CES" configuration task table in this chapter and refer to the *Cisco IOS Wide-Area Networking Configuration Guide* or the *OC-3/STM-1 ATM Circuit Emulation Service Network Module*.

To configure support for video streaming data over ATM AAL1 encapsulation using CES, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# network-clock base-rate {56k | 64k }`	Configures the network clock base rate. The keywords are as follows: • **56k**—Sets the network clock to 56 kbps. • **64k**—Sets the network clock to 64 kbps.
Step 2	`Router(config)# controller {t1 | e1} 0`	Selects T1/E1 controller 0. ATM is supported only on controller 0.
Step 3	`Router(config-controller)# mode atm`	Specifies that the controller will support ATM encapsulation and creates virtual ATM interface 0, which you will use to create the ATM permanent virtual circuits (PVCs).
Step 4	`Router(config-controller)# exit`	Exits controller configuration mode.

	Command	Purpose
Step 5	`Router(config)# interface atm 0 {multipoint | point-to-point}`	Enters interface configuration mode to configure ATM interface. • **0**—Indicates the ATM port number. Because the ATM interface processor (AIP) and all ATM port adapters have a single ATM interface, the port number is always 0. • **multipoint | point-to-point**—Specifies a multipoint or point-to-point subinterface.
Step 6	`Router(config-if)# pvc [name] vpi/vci [ilmi | qsaal | smds]`	Creates an ATM permanent virtual circuit (PVC) and enters ATM PVC configuration mode. The keywords and arguments are as follows: • *name*—(Optional) Specifies the name of the PVC or map. The name can be as many as 16 characters long. • *vpi*/—Specifies the ATM network VPI for the PVC that you named. The ATM network VPI of this PVC is an 8-bit field in the header of the ATM cell. The *vpi* value is unique only on a single link, not throughout the ATM network, because it has local significance only. The *vpi* value must match that of the switch. Valid values are from 0 to 255, but the value is usually 0 for ILMI communications. If a value is not specified, the *vpi* value is set to 0. **Note** You cannot set both *vpi* and *vci* to 0; if one is 0, the other cannot be 0. • *vci*—Specifies the ATM network VCI for the PVC you named. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has only local significance. The *vci* value ranges from 0 to 1 less than the maximum value set for this interface by the **atm vc-per-vp** command. **Note** Typically, the low *vci* values 0 to 31 are reserved for specific traffic (for example, F4 operations, administration, and maintenance [OAM]; SVC signaling; and ILMI). Do not use them for other PVCs.

	Command	Purpose
		• **ilmi**—(Optional) Sets up communication with the ILMI; the associated *vpi* and *vci* values ordinarily are 0 and 16, respectively. • **qsaal**—(Optional) Specifies a signaling-type PVC used for setting up or tearing down SVCs; the associated *vpi* and *vci* values ordinarily are 0 and 5, respectively. • **smds**—(Optional) A signaling-type PVC used for setting up or tearing down SVCs; the associated *vpi* and *vci* values ordinarily are 0 and 5, respectively.
Step 7	`Router(config-if-atm-pvc)# encapsulation aal1`	Sets the PVC to support ATM adaptation layer 1 (AAL1) encapsulation for video.
Step 8	`Router(config-if-atm-pvc)# cbr` *rate*	Configures the CBR for the ATM circuit emulation service (CES) for an ATM PVC on the Cisco MC3810 multiservice concentrator. By default, the *rate* argument used is the value configured with the **vc-class** command. The valid rate is from 56 to 10,000 kbps. The formula to calculate the CBR is 1.14 times the clock rate on the serial port.
Step 9	`Router(config-if-atm-pvc)# exit`	Exits ATM PVC configuration mode.
Step 10	`Router(config)# interface serial` *number* {**multipoint** | **point-to-point**}	Enters interface configuration mode for either serial port 0 or 1. For a full explanation of the keywords and argument, see Step 2 in the "Configuring Video in Pass-Through Mode" configuration task table in this chapter.
Step 11	`Router(config-if)# clock rate network-clock` *rate*	Configures the network clock speed for serial ports 0 or 1 in data circuit-terminating equipment (DCE) mode on the Cisco MC3810 multiservice concentrator. The *rate* argument is the network clock speed in bits per second. The range is from 56 kbps to 2048 kbps. The value entered should be a multiple of the value set for the **network-clock base-rate** command. The maximum rate supported is 2048 Mbps.
Step 12	`Router(config-if)# encapsulation atm-ces`	Enables CES ATM encapsulation on the Cisco MC3810.

	Command	Purpose
Step 13	`Router(config-if)# ces connect atm-interface pvc [name | [vpi/]vci]`	Maps the CES service to the PVC. The keywords and arguments are as follows: • *atm-interface*—Specifies the number of the ATM interface. The only valid option on the Cisco MC3810 multiservice concentrator is ATM0. • **pvc**—Specifies that the connection is to an ATM PVC. • *name*—(Optional) The name of the ATM PVC. • *vpi*/—(Optional) The virtual path identifier value. • *vci*—(Optional) The virtual channel identifier value.

Tuning Circuit Emulation Services Settings

Video streaming traffic over AAL1 uses CES. The default CES settings are sufficient for most configurations. However, you can tune the CES settings as needed.

To change the CES settings, use the following commands, beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ces initial-delay bytes`	Configures the maximum size of the CES circuit transmit buffer. The *bytes* argument specifies the size of the receive buffer of the CES circuit. The valid range is from 1 to 16,000 bytes. This command is used to accommodate cell jitter on the network. Bytes received from the ATM network are buffered by this amount before being sent to the CES port.
Step 2	`Router(config-if)# ces partial-fill octet`	Configures the number of user octets per cell for CES. The *octet* argument specifies the number of user octets per cell for the CES. Possible values of *octet* range from 0 to 47. Setting this number to zero disables partial cell fill and causes all cells to be completely filled before they are sent.

Configuring Video over ATM PVCs and SVCs

Video over ATM SVCs enables the Cisco MC3810 multiservice concentrator to provide dynamic and flexible videoconferencing system support. Using a plug-in VDM to provide an EIA/TIA-366 dialing interface to an H.320 video codec, the Cisco MC3810 automatically accepts dial-out requests from the video system. The codec connects to one of the Cisco MC3810 serial ports and also to the Cisco MC3810 EIA/TIA-366 dialup port.

This feature permits automatic PVC connections through a serial port. Each codec must place a call to the other videoconferencing system prior to the expiration of the video codec timeout period (set on the codec, usually 1 minute). Using a video dial map, each system reconciles the dialed number with a PVC that has already been configured, allowing fast connectivity.

This section describes the video functionality supported on the Cisco MC3810 and contains the following sections:

Service providers, educational organizations, and enterprises can combine video streams and packet data on a single high-speed ATM link. A separate ATM access multiplexer is not needed. Features of the Cisco ATM SVC implementation include the following:

- AAL1 and CES encapsulation is used to transport video traffic to the destination using a single CBR virtual circuit that includes multiple ATM SVCs.
- The implementation adheres to the required features of the ATM Forum UNI specification, version 4.0, which simultaneously supports PVCs and SVCs.
- Video over ATM SVCs support codec speeds of 128, 384, 768, and 1152 kbps.
- The Cisco MC3810, responding to the design of many leading H.320-based video systems, receives the called-party information from the EIA/TIA-366 interface and then reconciles the dialed address with a standard 20-octet ATM network service access point (NSAP) address.

Figure 137 shows a sample ATM video application.

Figure 137 Sample ATM Video Application

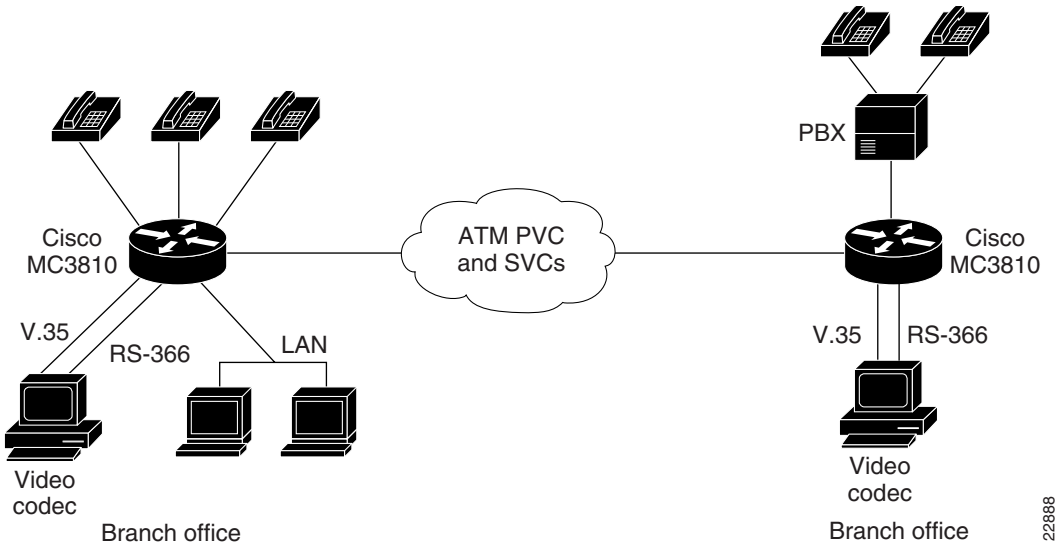
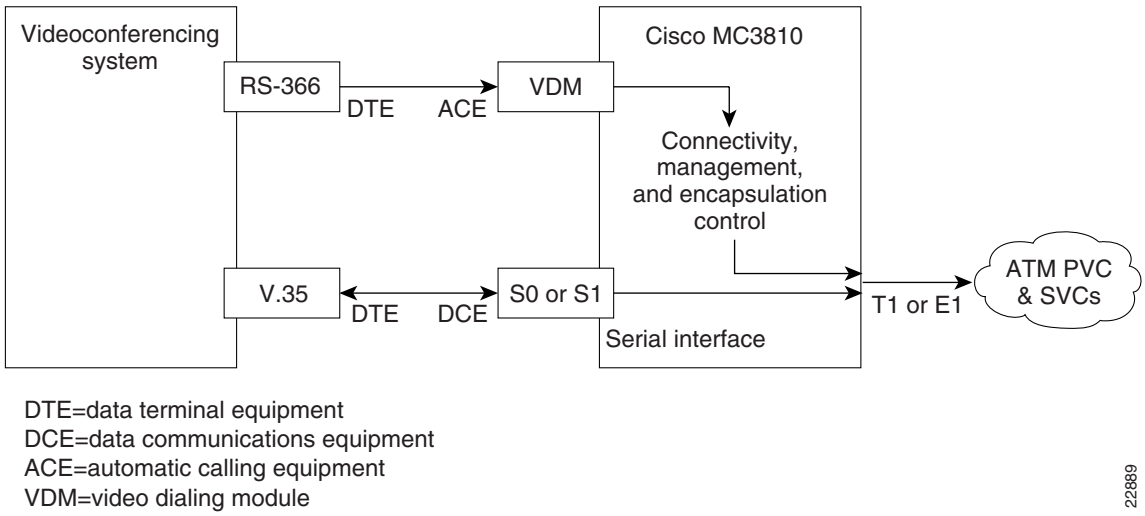


Figure 138 shows how the physical interfaces interact with software, the codec, and video data to handle connectivity and video functionality. The VDM automatic calling equipment (ACE) provides the EIA/TIA-366 interface to the video codec, and one of the Cisco MC3810 serial interfaces connects to the video codec DTE interface. The Video Call Manager (ViCM) software manages video calls that travel over a T1 or E1 facility through the Cisco MC3810 multiflex trunk (MFT) interface.

Figure 138 Physical Interfaces and Their Functions



The following restrictions apply to video over ATM using SVCs:

- Point-to-point connectivity for ATM SVC video does not support tandem switching and network (local) hunting.
- You can connect only one video codec to a Cisco MC3810.

- For video SVCs, the ATM service class is not configurable. It is automatically set to CBR, which is the standard service class for video.

The following special hardware is required for this feature:

- A Cisco MC3810 video dialing module VDM and an MFT module for ATM network connectivity
- Two cables:
  - A new Cisco serial V.35 DCE cable (product number 72-1721-01) that includes a ringing indicator (RI) conductor. This cable carries the video stream between the Cisco MC3810 and the video equipment. Videoconferencing equipment often uses the V.35 RI as the incoming call-alerting signal. Cisco standard serial V.35 cables do not include the RI conductor.
  - A Cisco EIA/TIA-366 ACE cable (product number 72-1722-01) to connect the VDM to the videoconferencing equipment EIA/TIA-366 dialup DTE port.

For additional information about installation and other hardware considerations, refer to the *Cisco MC3810 Multiservice Concentrator Hardware Installation Guide*.

Configuring Network Clocks and Controllers

Because real-time video communications require a continuous and tightly meshed data stream to avoid loss of information, you must synchronize source and destination devices to a single master clock. In the following example, the clock source is derived from a device attached to T1 controller 0; then it is distributed to the devices attached to the local Cisco MC3810 serial ports and to T1 controller 1. Clock source decisions should be based on the network configuration, and a hierarchy of clock sources can be set up so that backup clock sources are available. For details, see the "Configuring Synchronized Clocking" appendix.

To configure network clocks and the controller to support real-time video, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for controller T1/E1 0. The *number* argument indicates the network module number. The range is from 0 to 2. ATM traffic is supported on controller T1/E1 0 only.
Step 2	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 0 to obtain its clocking from the internal network clock PLL. (Use the **internal** keyword.) Keyword definitions are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed.

	Command	Purpose
		• **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as DCE clock sources. This situation assumes that the PBX also takes the clocking from the central office, thereby synchronizing the clocks on the DVM and the MFT.
Step 3	`Router(config-controller)# no shutdown`	Activates the controller.
Step 4	`Router(config)# controller {T1 | E1} 1`	Enters controller configuration mode for controller T1/E1 1. The *number* argument indicates the network module number. The range is from 0 to 2.
Step 5	`Router(config-controller)# clock source {line | internal | loop-timed}`	Sets the T1/E1 line clock source. For an explanation of the keywords, see Step 2 in this configuration task table.
Step 6	`Router(config-controller)# no shutdown`	Activates the controller.

Step 7	**T1 Line**
	`Router(config-controller)# framing {sf | esf}`
	E1 Line
	`Router(config-controller)# framing {crc4 | no-crc4} [australia]`
		Sets the framing for the E1 or T1 data line.
		The keywords are as follows:
		• **sf**—Specifies Super Frame as the T1 frame type.
		• **esf**—Specifies Extended Super Frame as the T1 frame type. This frame type is required for ATM on T1 lines. This setting is automatic for T1 when ATM mode is set.
		• **crc4**—Specifies CRC4 frame as the E1 frame type. This frame type is required for ATM on E1 lines. This setting is automatic for E1 when the ATM mode is set.
		• **no-crc4**—Specifies no CRC4 frame as the E1 frame type.
		• **australia**—(Optional) Specifies the E1 frame type used in Australia.

	Command	Purpose
Step 8	`Router(config-controller)# linecode {ami | b8zs | hdb3}`	Selects the line-code type for T1 or E1 lines. The keywords are as follows: • **ami**—Specifies alternate mark inversion (AMI) as the line-code type. It is valid for T1 or E1 controllers. This is the default for T1 lines. • **b8zs**—Specifies binary 8-zero substitution (B8ZS) as the line-code type. It is required for ATM on T1 lines. This setting is automatic for T1 when the ATM mode is set. • **hdb3**—Specifies high-density bipolar 3 (HDB3) as the line-code type. It is required for ATM on E1 lines. This setting is automatic for E1 when the ATM mode is set. **Note** When the E1 controller is specified, you must also configure scrambling on the ATM 0 interface. See Step 3 of the "Configuring ATM Interfaces to Support Video over PVCs and SVCs" configuration task table in this chapter.
Step 9	`Router(config-controller)# mode {atm | cas}`	Sets the mode of the T1/E1 controller and enters specific configuration commands for each mode type. The keywords are as follows: • **atm**—Sets the controller into ATM mode and creates an ATM interface (ATM 0) on the Cisco MC3810. When ATM mode is enabled, no channel groups, channel-associated signaling (CAS) groups, common channeling signaling (CCS) groups, or clear channels are allowed because ATM occupies all the DS0s on the T1/E1 trunk. When you set the controller to ATM mode, the controller framing is automatically set to ESF for T1 or CRC4 for E1. The line code is automatically set to B8ZS for T1 or HDBC for E1. When you remove ATM mode by entering the **no mode atm** command, ATM interface 0 is deleted. ATM mode is supported only on controller 0 (T1 or E1 0). • **cas**—Sets the controller into CAS mode, which allows you to create channel groups, CAS groups, and clear channels (both data and CAS modes). CAS mode is supported on both controllers 0 and 1.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode.

	Command	Purpose
Step 11	`Router(config)# network-clock base-rate {56k | 64k}`	Sets the network clock base rate for the serial ports. For video stream rates of 384, 768, 1.152, or 1.28 kbps, set the rate to 64 kbps. The default is 56 kbps. (see Step 1 in the "Configuring Video over ATM AAL1" configuration task table in this chapter.) **Note** At this point, you can also configure network protocol settings, such as IP hosts. For more information, see the *Cisco IOS IP Configuration Guide*.

Verifying Network Clock and Controller Configuration

To verify the configuration of network clock sources and controller settings, complete the following steps:

Step 1 Enter the **show network-clocks** privileged EXEC command to see the status of clock source settings. In this example, the "inactive config" clock setting is the current configuration:

```
Router# show network-clocks

Priority 1 clock source(inactive config): T1 0
Priority 1 clock source(active config): T1 0
Clock switch delay: 10
Clock restore delay: 10
T1 0 is clocking system bus for 9319 seconds.
Run Priority Queue: controller0
```

Step 2 Enter the **show controllers t1** or **show controllers e1** privileged EXEC commands to see the status of T1 or E1 controllers, as in the following example:

```
Router# show controller t1 1

T1 1 is up.
  Applique type is Channelized T1
  Cablelength is long gain36 0db
  No alarms detected.
  Slot 4 CSU Serial #07789650 Model TEB HWVersion 4.70 RX level = 0DB
  Framing is ESF, Line Code is B8ZS, Clock Source is Internal.
  Data in current interval (819 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Data in Interval 1:
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Data in Interval 2:
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  .
  .
  .
  Data in Interval 96:
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

```
  Total Data (last 24 hours)
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs

!
Router# show controllers E1 1
E1 1 is up.
  Applique type is Channelized E1 - balanced
  No alarms detected.
  Slot 4 Serial #06868949 Model TEB HWVersion 3.80
  Framing is CRC4, Line Code is HDB3, Clock Source is Internal.
  Data in current interval (292 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
    .
    .
    .
  Total Data (last 66 15 minute intervals):
     9 Line Code Violations, 0 Path Code Violations,
     1 Slip Secs, 0 Fr Loss Secs, 4 Line Err Secs, 0 Degraded Mins,
     5 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
```

Configuring Serial Interfaces to Support the Video Codec

The configuration of serial interfaces to support the video codec is supported only on the Cisco MC3810 multiservice concentrator.

To configure the serial interfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial {0 | 1}`	Enters interface configuration mode for either for serial 0 or serial 1, depending on where the video codec is connected.
Step 2	`Router(config-if)# clock rate network` *rate*	Configures the network clock speed for DCE mode, in bits per second, corresponding to the video stream rate you are using. The *rate* must be a multiple of the value set with the **network-clock base-rate** command in Step 11 of the "Configuring Network Clocks and Controllers" configuration task table in this chapter. Make sure this setting is 384000, 768000, or 1152000. 768000 is a common setting.
Step 3	`Router(config-if)# encapsulation atm-ces`	Configures the interface for ATM encapsulation circuit emulation service (CES), which is required for video codec support.

	Command	Purpose
Step 4	`Router(config-if)# serial restart-delay count`	Sets the amount of time that the router waits before trying to bring up a serial interface when the interface goes down. The router resets the hardware each time the restart timer expires. This command is often used with dial backup and with the **pulse-time** command, which sets the amount of time to wait before redialing when a data terminal ready (DTR) dialed device fails to connect. The *count* argument is a value from 0 to 900 in seconds. This is the frequency at which the hardware is reset. A value of 0 means that the hardware is not reset when down. If the interface is used to answer a call, it does not cause the DTR circuit to drop. If the DTR circuit drops, the modem can disconnect.

Verifying Serial Interface Configuration for Video Codecs

To see the status of all serial interfaces or of a specific serial interface, enter the privileged EXEC command **show interfaces serial** as shown in the example below. You can use this command to check the encapsulation, scrambling, and serial restart delay settings:

```
Router# show interface serial0

Serial0 is down, line protocol is down
  Hardware is PQUICC Serial Trans
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
     reliability 255/255, txload 65/255, rxload 1/255
  Encapsulation CES-ATM, loopback not set
  Keepalive not set
  Scramble enabled
  Restart-Delay is 0 secs
  Last input never, output never, output hang never
  Last clearing of "showshow interface" counters 5d13h
  Queueing strategy: fifo
  Output queue 0/100, 101 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     13452224 packets input, 1526136219 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     215189699 packets output, 1654453088 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
  Cable attached: V.35 (DCE)
  Hardware config: V.35; DCE; PLL nx64K;
  DSR = UP   DTR = DOWN   RTS = DOWN   CTS = DOWN   DCD = DOWN
```

Configuring ATM Interfaces to Support Video over PVCs and SVCs

This section demonstrates how to set up the ATM interface and how to configure the ATM interface to support video over PVCs and SVCs. The video NSAP addressing commands specify session target information for SVC video communications.

This feature is supported only on the Cisco MC3810 multiservice concentrator.

To configure ATM interfaces to support video over PVCs and SVCs (including configuring a dial PVC for videoconferencing), use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface atm` *slot*`/`*port*`.`*subinterface-number* {**multipoint** | **point-to-point**}	Enters interface configuration mode. The keywords and arguments are as follows: • *slot*—Specifies the backplane slot number on your router. The value ranges from 0 to 4, depending on what router you are configuring. Refer to your router hardware documentation. • **/0**—ATM port number. Because the ATM Interface Processor (AIP) and all ATM port adapters have a single ATM interface, the port number is always 0. • .*subinterface-number*—Specifies a subinterface number in the range from 1 to 4294967293. • **multipoint**—Specifies that your network is fully meshed and you want to communicate with multiple routers. • **point-to-point**—Configures the subinterface for communication with one router, as in a hard-wired connection. There is no default for this parameter.
Step 2	`Router(config-if)# ip address` *ip-address mask* [**secondary**]	For IP protocol communications, assigns the IP address and subnet mask to the interface. The keywords and arguments are as follows: • *ip-address*—IP address. • *mask*—Mask for the associated IP subnet. • **secondary**—(Optional) Specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 3	`Router(config-if)# atm scramble-enable`	(E1 configuration only) Helping to ensure reliability, scrambling randomizes the ATM cell payload frames to avoid continuous nonvariable bit patterns and to improve the efficiency of ATM cell delineation algorithms.

	Command	Purpose
Step 4	`Router(config-if)# atm video aesa {default |esi-address}`	Sets the unique ATM end-station address (AESA) for an ATM video interface that is using switched virtual circuit (SVC) mode. The keywords and arguments are as follows: • **default**—Automatically creates a network service access point (NSAP) address for the interface, based on a prefix from the ATM switch (26 hexadecimal characters), the MAC address (12 hexadecimal characters) as the end system identifier (ESI), and a selector byte (two hexadecimal characters). • *esi-address*—Requires that you enter 12 hexadecimal characters as the ESI. The ATM switch provides the prefix, and the video selector byte provides the remaining characters. You can view the assigned address by using the **show atm video-voice address** command.
Step 5	`Router(config-if)# pvc [name] vpi/vci [ilmi | qsaal | smds]`	Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters interface-ATM-VC configuration mode. **Note** To set up communication with the Integrated Local Management Interface (ILMI), enter the **ilmi** keyword for ATM adaptation layer encapsulation; the associated *vpi* and *vci* values are ordinarily 0 and 16, respectively. **Note** To enable the signaling for setup and teardown of SVCs, specify the Q.SAAL (signaling ATM adaptation layer) encapsulation as the *name*; the associated *vpi* and *vci* values are ordinarily 0 and 5, respectively. You cannot create this PVC on a subinterface.

Command	Purpose
	Complete keyword and argument definitions are as follows: • *name*—(Optional) Specifies a unique label that can be up to 16 characters long. It identifies to the processor the virtual path identifier-virtual channel identifier (VPI-VCI) pair to use for a particular packet. • *vpi*/—Specifies the ATM network VPI for the PVC that you named. The absence of the "/" and a vpi value defaults the vpi value to 0. The ATM network VPI of this PVC is an 8-bit field in the header of the ATM cell. The *vpi* value is unique only on a single link, not throughout the ATM network, because it has local significance only. The *vpi* value must match that of the switch. Valid values are from 0 to 255, but the value is usually 0 for ILMI communications. If a value is not specified, the *vpi* value is set to 0. **Note** You cannot set both *vpi* and *vci* to 0; if one is 0, the other cannot be 0. • *vci*—Specifies the ATM network VCI for the PVC you named. The VCI is a 16-bit field in the header of the ATM cell. The VCI value is unique only on a single link, not throughout the ATM network, because it has only local significance. The *vci* value ranges from 0 to 1 less than the maximum value set for this interface by the **atm vc-per-vp** command. **Note** Typically, the low *vci* values 0 to 31 are reserved for specific traffic (for example, F4 operations, administration, and maintenance [OAM]; SVC signaling; and ILMI). Do not use them for other PVCs. • **ilmi**—(Optional) Sets up communication with the ILMI; the associated vpi and vci values ordinarily are 0 and 16, respectively.

	Command	Purpose
		• **qsaal**—(Optional) Specifies a signaling-type PVC used for setting up or tearing down SVCs; the associated *vpi* and *vci* values ordinarily are 0 and 5, respectively. • **smds**—(Optional) Specifies encapsulation for switched multimegabit data service (SMDS) networks. If you are configuring an ATM PVC on the ATM Interface Processor (AIP), you must configure AAL3/4SMDS using the **atm aal aal3/4** command before specifying **smds** encapsulation. If you are configuring an ATM network processor module (NPM), the **atm aal aal3/4** command is not required. SMDS encapsulation is not supported on the ATM port adapter.
Step 6	`Router(config-if-atm-pvc)# protocol` *protocol* {*protocol-address* | **inarp**} [[**no**] **broadcast**]	Configures a static map for an ATM permanent PVC, SVC, or virtual circuit (VC) class or enables Inverse Address Resolution Protocol (ARP) or Inverse ARP broadcasts on an ATM PVC. The keywords and arguments are as follows: • *protocol*—Specifies one of the following: – **aarp**—AppleTalk ARP – **apollo**—Apollo domain – **appletalk**—AppleTalk – **arp**—IP ARP**bridge**—bridging – **bstun**—block serial tunnel – **cdp**—Cisco Discovery Protocol – **clns**—ISO Connectionless Network Service (CLNS) – **clns_es**—ISO CLNS end system – **clns_is**—ISO CLNS intermediate system**cmns**—ISO Connection-Mode Network Service (CMNS) – **compressedtcp**—Compressed TCP – **decnet**—DECnet**decnet_node**—DECnet node – **decnet_prime_router**—DECnet prime router – **decnet_router-l1**—DECnet router L1 – **decnet_router-l2**—DECnet router L2

Command	Purpose
	– **dlsw**—data link switchin – **ip**—IP – **ipx**—Novell IPX – **llc2**—llc2 – **pad**—Packet assembler/disassembler (PAD) links – **qllc**—Qualified Logical Link Control protocol – **rsrb**—remote source-route bridging – **snapshot**—snapshot routing support – **stun**—serial tunnel – **vines**—Banyan VINES – **xns**—Xerox Network Systems protocol • *protocol-address*—Specifies the destination address that is being mapped to a PVC. • **inarp**—(Valid only for IP and IPX protocols on PVCs) Enables Inverse ARP on an ATM PVC. If you specify *protocol-address* instead of **inarp**, Inverse ARP is automatically disabled for that protocol. • [**no**] **broadcast**—(Optional) broadcast indicates that this map entry is used when the corresponding protocol sends broadcast packets to the interface (for example, Interior Gateway Routing Protocol [IGRP] updates. Pseudobroadcasting is supported. The broadcast keyword of the protocol command takes precedence if you previously configured the broadcast command on the ATM PVC or SVC.
Step 7 `Router(config-if-atm-pvc)# cbr rate`	Configures the CBR for the ATM circuit emulation service (CES) for an ATM PVC on the Cisco MC3810 multiservice concentrator. By default, the *rate* argument used is the value configured with the **vc-class** command. The valid rate is from 56 to 10,000 kbps. The formula for calculating the CBR is 1.14 times the clock rate on the serial port.
Step 8 `Router(config-if-atm-pvc)# encapsulation aal1`	Configures ATM adaptation layer 1 (AAL1) encapsulation necessary for videoconferencing using PVCs.

Verifying ATM Interface Configuration for Video over PVCs and SVCs

To verify ATM interface configuration, complete the following steps:

Step 1 Enter the **show atm pvc** command with the VPI/VCI specified to see the PVCs that are set up for ILMI management and Q.SAAL signaling, as in the following examples:

```
Router# show atm pvc 0/5

ATM0: VCD: 2, VPI: 0, VCI: 5, Connection Name: SAAL
UBR, PeakRate: 56
AAL5-SAAL, etype:0x4, Flags: 0x26, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s), OAM retry frequency: 1
second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not Managed
ILMI VC state: Not Managed
InARP DISABLED
InPkts: 2044, OutPkts: 2064, InBytes: 20412, OutBytes: 20580
InPRoc: 2044, OutPRoc: 2064, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Compress: Disabled
Status: INACTIVE, State: NOT_IN_SERVICE
!
Router# show atm pvc 0/16

ATM0: VCD: 1, VPI: 0, VCI: 16, Connection Name: ILMI
UBR, PeakRate: 56
AAL5-ILMI, etype:0x0, Flags: 0x27, VCmode: 0x0
OAM frequency: 0 second(s), OAM retry frequency: 1 second(s), OAM retry frequency: 1
second(s)
OAM up retry count: 3, OAM down retry count: 5
OAM Loopback status: OAM Disabled
OAM VC state: Not Managed
ILMI VC state: Not Managed
InARP DISABLED
InPkts: 398, OutPkts: 421, InBytes: 30493, OutBytes: 27227
InPRoc: 398, OutPRoc: 421, Broadcasts: 0
InFast: 0, OutFast: 0, InAS: 0, OutAS: 0
OAM cells received: 0
F5 InEndloop: 0, F5 InSegloop: 0, F5 InAIS: 0, F5 InRDI: 0
F4 InEndloop: 0, F4 InSegloop: 0, F4 InAIS: 0, F4 InRDI: 0
OAM cells sent: 0
F5 OutEndloop: 0, F5 OutSegloop: 0, F5 OutRDI: 0
F4 OutEndloop: 0, F4 OutSegloop: 0, F4 OutRDI: 0
OAM cell drops: 0
Compress: Disabled
Status: INACTIVE, State: NOT_IN_SERVICE
```

Step 2 Enter the **show interface atm 0** privileged EXEC command to see information about the ATM interface, as in the following example:

```
Router# show interface atm 0

ATM0 is up, line protocol is up
  Hardware is PQUICC Atom1
  Internet address is 9.1.1.6/8
  MTU 1500 bytes, sub MTU 1500, BW 1536 Kbit, DLY 20000 usec,
     reliability 255/255, txload 22/255, rxload 11/255
  NSAP address: 47.0091810000000002F26D4901.000011116666.06
  Encapsulation ATM
  292553397 packets input, 3437519137 bytes
  164906758 packets output, 1937663833 bytes
  0 OAM cells input, 0 OAM cells output, loopback not set
  Keepalive not supported
  Encapsulation(s):, PVC mode
  1024 maximum active VCs, 28 current VCCs
  VC idle disconnect time: 300 seconds
  Signalling vc = 1, vpi = 0, vci = 5
  UNI Version = 4.0, Link Side = user
  Last input 00:00:00, output 2d05h, output hang never
  Last clearing of "show interface" counters never
  Input queue: -1902/75/0 (size/max/drops); Total output drops: 205
  Queueing strategy: weighted fair
  Output queue: 0/1000/64/0 (size/max total/threshold/drops)
     Conversations  0/0/256 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 67000 bits/sec, 273 packets/sec
  5 minute output rate 136000 bits/sec, 548 packets/sec
     76766014 packets input, 936995443 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     367264676 packets output, 3261882795 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Step 3 Enter the **show atm vc** privileged EXEC command to see how SVCs and PVCs are set up, as in the following example:

```
Router# show atm vc

 VCD /                                        Peak  Avg/Min Burst
Interface   Name       VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
0           1           0     5   PVC    SAAL     UBR        56               UP
0           2           0    16   PVC    ILMI     UBR        56               UP
0           3           34   35   PVC    AAL1     CBR  768    768              UP
0           4           38   39   SVC     CES     CBR  768    768              UP
```

Step 4 Enter the **show atm video-voice address** privileged EXEC command to see information about the ATM interface address, which is particularly helpful because the address is assigned automatically through the **atm voice aesa** command. The following example also confirms that the ILMI status is confirmed—the ILMI PVC is set up to allow SVC management:

```
Router# show atm video-voice address

nsap address                                    type         ilmi status
47.0091810000000002F26D4901.00107B4832E1.FE   VOICE_AAL5   Confirmed
47.0091810000000002F26D4901.00107B4832E1.C8   VIDEO_AAL1   Confirmed
```

Configuring Video Dial Peers

The video dial peer feature is supported on only the Cisco MC3810 multiservice concentrator.

To configure video dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# port signal slot/port`	Specifies the slot where the video dialing module (VDM) is located and the port for the EIA/TIA-366 interface. The arguments are as follows: • *slot*—Indicates that the value of the VDM is either 1 or 2. • *port*—Enters the port location of the RS-366 interface. The Cisco MC3810 VDM has only one video port, so the *port* value is 0.
Step 2	`Router(config)# dial-peer video tag {videocodec | videoatm}`	Defines a video ATM dial peer for the remote system and enters dial-peer configuration mode. Video dial peers are persistent and exist until they are specifically removed with the **no** form of the **dial-peer video** command. The keywords and arguments are as follows: • *tag*—Identifies the dial peer and must be unique on the Cisco MC3810. Do not duplicate a specific tag number. Valid values are from 1 to 10000. • **videocodec**—Specifies a local video codec connected to the router. • **videoatm**—Specifies a remote video codec on the ATM network.
Step 3	`Router(config-dial-peer)# destination-pattern [+] string [T]`	Specifies the E.164 address associated with this dial peer. The keywords and arguments are as follows: • **+**—(Optional) Specifies a character indicating an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810. • *string*—Indicates a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – The asterisk (*) and pound sign (#)—Indicate the keys that appear on standard touch-tone dial pads. On the Cisco 3600 series only, these characters cannot be used as leading characters in a string (for example, *650).

Command	Purpose
	– Comma (,)—Inserts a pause between digits. – Period (.)—Matches any entered digit (this character is used as a wildcard). On the Cisco 3600 series, the period cannot be used as a leading character in a string (for example, .650). – Percent sign (%)—Indicates that the preceding digit or pattern occurred zero or multiple times, similar to the wildcard in the regular expression. – Plus sign (+)—Specifies a sequence of one or more of the character or pattern. **Note** The plus sign used as part of the digit string is different from the plus sign that can be used in front of the digit string to indicate that the string is an E.164 standard number. – Circumflex accent (^)—Indicates a match to the beginning of the string. – Dollar sign ($)—Matches the null string at the end of the input string. – Backslash symbol (\)—Indicates a character followed by a single character matching the first character or by a single character having no other significance. – Question mark (?)— Indicates that the preceding digit occurred zero or one time. – Brackets ([])—Indicate a range of digits. A range is a sequence of characters enclosed in the brackets; only numeric characters from 0 to 9 are allowed in the range. This is similar to a regular expression rule. – Parentheses ( )—Indicate a pattern and are the same as the regular expression rule, for example, 408(555). Parentheses are used in conjunction with symbols ?, %, and +. For more information on applying wildcard symbols to destination patterns and the dial strings that result, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter. • **T**—(Optional) Control character indicating that the **destination-pattern** value is a variable-length dial string.

	Command	Purpose
Step 4	**Cisco MC3810 Multiservice Concentrator** `Router(config-dial-peer)# session target {serial | atm} interface {svc nsap nsap-address | pvc {name| vpi/vci | vci}`	Configures the ATM session target for the dial peer. The keywords and arguments are as follows: • **serial**—Specifies the serial interface for the dial-peer address. • **atm**—Specifies the ATM interface number. The only valid number is 0. • *interface*—Specifies the interface number. • **svc nsap**—Specifies the switched virtual circuit (SVC) network service access point (NSAP) address. • *nsap-address*—Specifies a 40-digit hexadecimal number for the session target NSAP. • **pvc**—Specifies a permanent virtual circuit (pvc). • *name*—Specifies the name of the session target ATM PVC. • *vpi/vci*—Specifies the ATM network virtual path identifier (VPI) and virtual channel identifier (VCI) of this PVC. • *vci*—Specifies the ATM network VCI of this PVC. **Note** If you are using PVCs to send video data, you can also specify a PVC defined on the ATM interface as a session target by using a name or a VPI-VCI combination.
Step 5	`Router(config-dial-peer)# exit`	Exits dial peer configuration mode.
Step 6	`Router(config)# dial-peer video tag {videocodec | videoatm}`	Defines a video ATM dial peer for the local video codec. The keywords and arguments are as follows: • *tag*—Defines the dial peer and assigns the protocol type to the peer. Valid entries are from 1 to 10000. The tag must be unique on the router. • **videocodec**—Specifies a local video codec connected to the router. • **videoatm**—Specifies a remote video codec on the ATM network.
Step 7	`Router(config-dial-peer)# destination-pattern [+] string [T]`	Specifies the E.164 address associated with this dial peer. For an explanation of the keywords and arguments, see Step 3 of this configuration task table.
Step 8	`Router(config-dial-peer)# port signal slot/port`	Specifies the slot where the VDM is located and the port for the EIA/TIA-366 interface. For an explanation of the arguments, see Step 1 in this configuration task table.

	Command	Purpose
Step 9	`Router(config-dial-peer)# port media` *interface*	Specifies the serial interface where the local video codec is connected for a local video dial peer. The *interface* argument indicates the serial interface where the local codec is connected. Valid entries are the numbers 1 or 0.
Step 10	`Router(config-dial-peer)# nsap` *nsap-address*	Specifies the NSAP address for the codec. The *nsap-address* argument is a 40-digit hexadecimal number that must be unique on the device.

Verifying Video Dial-Peer Configuration

To verify the dial-peer configuration, enter the **show dial-peer video** privileged EXEC command. In the following example, note that the third dial peer uses a PVC specified with a VPI-VCI value while the second uses an SVC. The first dial peer is for the local codec.

```
Router# show dial-peer video

Video Dial-Peer 1
      type = videocodec, destination-pattern = 111
      port signal = 1/0, port media = Serial1
      nsap = 47.0091810000000050E201B101.00107B09C6F2.C8
Video Dial-Peer 2
      type = videoatm,   destination-pattern = 222
      session-target = ATM0 svc nsap 47.0091810000000050E201B101.00E01E92ADC2.C8
Video Dial-Peer 3
      type = videoatm,   destination-pattern = 333
      session-target = ATM0 pvc 70/70
```

Troubleshooting Video over ATM SVCs and PVCs

When problems occur with video over ATM PVCs or SVCs on the Cisco MC3810 multiservice concentrator, perform the following steps to find the source of your problems. Common problems are addressed before more complex problems:

Note If you are using dial PVCs (rather than SVCs) for video communications, ensure that both parties dial one another within the timeout period that is set on the codec. This timeout period is usually one minute.

Step 1 Check the LEDs on the EIA/TIA-366 interface. If the green LED is not lit, there may be a hardware problem, or the correct image may not be loaded. For more information, see the *Cisco MC3810 Multiservice Concentrator Hardware Installation Guide*.

Step 2 Make sure that the ATM interface, serial ports, and controllers are set to **no shutdown**.

Step 3 Check the serial interface configuration.

- If you are using dial PVCs for video, do not include the **ces connect** serial interface command because this command does not provide mapping to the ATM interface for PVCs (or SVCs) for the dial video feature. Instead, create dial PVCs under ATM interface configuration. If the **ces connect** command has been configured, it appears in **show running-config** command output under serial interface 0 or 1.
- Enter the **show interfaces serial** privileged EXEC command. Ensure that the serial interface communications circuitry is operational, as shown in the last line of the **show interfaces serial** command output:

```
DSR = UP   DTR = UP   RTS = UP   CTS = UP   DCD = UP
```

Step 4 (For SVCs only) On both Cisco MC3810 multiservice concentrators, make sure that ILMI and Q.SAAL PVCs are set up to allow SVC communications. The **show atm pvc** privileged EXEC command displays information about configured PVCs, including the ILMI and Q.SAAL PVCs.

```
Router# show atm pvc

 VCD /                                        Peak  Avg/Min Burst
Interface   Name       VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
0           1           0     5   PVC    SAAL     UBR       56              UP
0           2           0    16   PVC    ILMI     UBR       56              UP
```

Step 5 (For dial PVCs only) On both Cisco MC3810 multiservice concentrators, make sure that PVCs are set up to allow dial PVC connections and that CBR is the configured service class (SC). In addition, the bit rate must correspond to the rate set on the serial interface. The **show atm pvc** privileged EXEC command displays information about configured PVCs.

```
Router# show atm vc

 VCD /                                        Peak  Avg/Min Burst
Interface   Name       VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells  Sts
0           3           38   35   PVC    AAL1     CBR  384    384             UP
```

Step 6 (For SVCs only) Ensure that NSAP addresses are set up and confirmed as operational under the ATM interfaces of the Cisco MC3810 multiservice concentrators on both sides of the communication. Enter the **show atm video-voice address** or **show atm ilmi-status** privileged EXEC commands, as shown in the following example. The **show atm ilmi-status** command provides more details about the ILMI PVC than does the **show atm video-voice address** command.

```
Router# show atm video-voice address

nsap address                                    type         ilmi status
47.0091810000000002F26D4901.00107B4832E1.FE   VOICE_AAL5   Confirmed
47.0091810000000002F26D4901.00107B4832E1.C8   VIDEO_AAL1   Confirmed

Router# show atm ilmi-status

Interface : ATM0 Interface Type : Private UNI (User-side)
ILMI VCC : (0, 16) ILMI Keepalive : Enabled (5 Sec 4 Retries)
ILMI State:       UpAndNormal
Peer IP Addr:     10.1.1.11       Peer IF Name:     ATM1/0/0
Peer MaxVPIbits:  8               Peer MaxVCIbits:  14
Active Prefix(s) :
47.0091.8100.0000.0002.f26d.4901
End-System Registered Address(s) :
47.0091.8100.0000.0002.f26d.4901.0000.1111.5555.05(Confirmed)
47.0091.8100.0000.0002.f26d.4901.0010.7b48.32e1.fe(Confirmed)
47.0091.8100.0000.0002.f26d.4901.0010.7b48.32e1.c8(Confirmed)
```

Step 7 Check for clocking problems. Enter the **show controllers t1** or **show controllers e1** privileged EXEC command to check for slip errors, as shown in the following excerpt from the command output:

```
 .
 .
 .
Data in current interval (819 seconds elapsed):
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Data in Interval 1:
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Data in Interval 2:
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
  Data in Interval 3:
     0 Line Code Violations, 0 Path Code Violations
     0 Slip Secs, 0 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 0 Unavail Secs
    .
    .
    .
```

A few slip errors may not indicate a problem with clocking. However, if there are numerous errors, check the following possibilities:

- The network clocks are not set to the same clock rate. Enter the **show network-clocks** command on the devices to ensure that these clock rates match.
- The Cisco MC3810 multiservice concentrators may not be using the same clock source. For example, if there are two back-to-back Cisco MC3810 multiservice concentrators and one is using an internal clock source, the other must use the line clock source to obtain clocking from the same device. Enter the **show network-clocks** and **show controllers t1** or **show controllers e1** commands to see the clock source settings. For additional guidance, see the "Configuring Synchronized Clocking" appendix.

Step 8 Check the functionality of the Service-Specific Connection-Oriented Protocol (SSCOP). Enter the **show sscop** privileged EXEC command. See the following excerpt from the command output:

```
Router# show sscop

SSCOP details for interface ATM0
   Current State = Data Transfer Ready
```

Interpretation of the command output requires familiarity with SSCOP, so unless you are familiar with the protocol, just use the command to ensure that the protocol is in a state of readiness. If you need to make changes, see the *Cisco IOS Wide-Area Networking Configuration Guide*.

Note If you plan to adjust SSCOP parameters, you may wish to complete the rest of the troubleshooting steps before doing so.

Step 9 Enter the **show dial-peer video** command on the local and remote concentrators to verify that each has been configured properly to communicate with the other:

```
Router1# show dial-peer video

dial-peer video 111 videocodec
 nsap 47.0091810000000002F26D4901.00107B4832E1.C8
 port signal 1/0
```

```
 port media Serial0
 destination-pattern 121
!
dial-peer video 221 videoatm
 destination-pattern 221
 session target ATM0 svc nsap 47.0091810000000002F26D4901.00107B09C645.C8

Router2# show dial-peer video

dial-peer video 111 videocodec
 nsap 47.0091810000000002F26D4901.00107B09C645.C8
 port signal 1/0
 port media Serial0
 destination-pattern 221
!
dial-peer video 121 videoatm
 destination-pattern 121
 session target ATM0 svc nsap 47.0091810000000002F26D4901.00107B4832E1.C8
```

Step 10 Enter the **show video call summary** command to quickly check the status of calls on the local and remote multiservice access concentrators. "ViCM" is the internal video call manager.

When no call is in progress, the output looks like this:

```
Router# show video call summary

Serial0:ViCM = Idle, Codec Ready
```

When a call is starting, the output looks like this:

```
Router# show video call summary

Serial0:ViCM = Call Connected
```

When a call is disconnecting, the output looks like this:

```
Router# show video call summary

Serial0:ViCM = Idle
```

Step 11 Enter the privileged EXEC **show call history video record** command to see information about current and recent video calls, allowing analysis of possible problems:

```
Router# show call history video record

CallId = 4
CalledNumber = 221
CallDuration = n/a - call is in progress
DisconnectText = n/a - call is in progress
SVC: call ID = 8598630
Remote NSAP = 47.0091810000000002F26D4901.00107B09C645.C8
Local NSAP = 47.0091810000000002F26D4901.00107B4832E1.C8
vcd = 414, vpi = 0, vci = 158
SerialPort = Serial0
VideoSlot = 1, VideoPort = 0


CallId = 3
CalledNumber = 221
CallDuration = 557 seconds
DisconnectText = local hangup
SVC: call ID = 8598581
Remote NSAP = 47.0091810000000002F26D4901.00107B09C645.C8
Local NSAP = 47.0091810000000002F26D4901.00107B4832E1.C8
vcd = 364, vpi = 0, vci = 108
```

```
SerialPort = Serial0
VideoSlot = 1, VideoPort = 0


CallId = 2
CalledNumber = n/a - incoming call
CallDuration = 125 seconds
DisconnectText = local hangup
SVC: call ID = 8598484
Remote NSAP = n/a
Local NSAP = 47.0091810000000002F26D4901.00107B4832E1.C8
vcd = 264, vpi = 0, vci = 273
SerialPort = Serial0
VideoSlot = 1, VideoPort = 0


CallId = 1
CalledNumber = n/a - incoming call
CallDuration = 171651 seconds
DisconnectText = remote hangup
SVC: call ID = 8594356
Remote NSAP = n/a
Local NSAP = 47.0091810000000002F26D4901.00107B4832E1.C8
vcd = 7, vpi = 0, vci = 39
SerialPort = Serial0
VideoSlot = 1, VideoPort = 0
```

Step 12 Enter the **debug video vicm** command to follow in-progress calls carefully. Comments are framed in asterisks (*):

```
Router# debug video vicm

Video ViCM FSM debugging is on

***** Starting Video call *****

Router# SVC HANDLE in rcvd:0x80001B:

00:42:55:ViCM - current state = Idle, Codec Ready
00:42:55:ViCM - current event = SVC Setup
00:42:55:ViCM - new state = Call Connected

00:42:55:ViCM - current state = Call Connected
00:42:55:ViCM - current event = SVC Connect Ack
00:42:55:ViCM - new state = Call Connected

*****Video Call Disconnecting*****

Router#

00:43:54:ViCM - current state = Call Connected
00:43:54:ViCM - current event = SVC Release
00:43:54:ViCM - new state = Remote Hangup

00:43:54:ViCM - current state = Remote Hangup
00:43:54:ViCM - current event = SVC Release Complete
00:43:54:ViCM - new state = Remote Hangup
mc3810_video_lw_periodic:Codec is not ready
mc3810_video_lw_periodic:sending message

00:43:55:ViCM - current state = Remote Hangup
00:43:55:ViCM - current event = DTR Deasserted
00:43:55:ViCM - new state = Idle
```

```
mc3810_video_lw_periodic:Codec is ready
mc3810_video_lw_periodic:sending message

00:43:55:ViCM - current state = Idle
00:43:55:ViCM - current event = DTR Asserted
00:43:55:ViCM - new state = Idle, Codec Ready
```

Configuring the CES Clock

The OC-3/STM-1 ATM CES network module uses the CES clock and passes the clocking information to the T1 controller and to the ATM interface. The clock must be set up on the CES interface, and then the T1 controller and ATM interface must be configured to use either its own physical loop or the clocking information that is passed. Some examples of the CES clock settings are shown at the end of this section.

To configure video support over ATM AAL1 PVCs, it is also necessary to perform the tasks in the "Configuring Structured CES" configuration task table in this chapter.

This feature is supported on the Cisco 3600 series routers.

To configure the CES clock, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ces` *slot/port*	Configures Circuit Emulation Service (CES) on a router port and enters CES configuration mode. The *slot/port* argument indicates the backplane slot number and port number on the interface. The port value is always 0 because the interface configuration applies to all ports in the slot.
Step 2	`Router(config-ces)# clock-select` *priority-no interface slot/port*	Establishes the sources and priorities of the requisite clocking signals for the OC-3/STM-1 ATM CES network module. The arguments are as follows: • *priority-no*—Indicates the priority of the clock source. The values are from 1 (high priority) to 4 (low priority). • *interface*—Specifies the interface that will supply the clock source. • *slot/port*—Specifies the backplane slot number and port number on the interface.
Step 3	`Router(config-ces)# exit`	Exits CES configuration mode.
Step 4	`Router(config)# controller {T1 | E1}` *slot/port*	Enters controller configuration mode for the T1 or E1 controller at the specified *slot/port* location. The prompt changes again to show that you are in controller configuration mode.

	Command	Purpose
Step 5	`Router(config-controller)# clock source {line {primary | secondary} | internal}`	Specifies which end of the circuit provides clocking for the T1 or E1 interface. The keywords are as follows: • **line**—Specifies that the interface will clock its transmitted data from a clock recovered from the line's receive data stream. This is the default. • **primary**—Specifies the source of primary line clocking. The default primary TDM clock source is from the T0 controller. • **secondary**—Specifies the source of secondary line clocking. The default secondary TDM clock source is from the T1 controller. • **internal**—Specifies that the interface will clock its transmitted data from its internal clock. **Note** The clock source should be set to use internal clocking when the installed video WAN interface card (VWIC) uses the clocking designated by the CES clock setting.
Step 6	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 7	`Router(config)# interface atm` *slot/port*	Configures the clocking on the ATM interface and enters interface configuration mode. The *slot/port* arguments are as follows: • *slot*—Specifies the backplane slot number on the router. The value ranges from 0 to 4, depending on what router you are configuring. Refer to your router hardware documentation. • *port*—Specifies the port number. The number depends on the number of ports on the network module.
Step 8	`Router(config-if)# atm clock internal`	Specifies which end of the circuit provides clocking for the ATM interface. The clock source should be set to use internal clocking when the CES clock is set to anything other than ATM. The **no atm clock internal** command should be set if using the ATM physical loop for clocking

Configuring Structured CES

Structured CES allows you to allocate bandwidth in a highly flexible and efficient manner. With structured services, you use only the bandwidth actually required to support the active structured circuits that you configure.

Structured CES is supported on Cisco 3600 series routers for video over AAL1 using the OC-3/STM-1 ATM CES network module.

For information on configuring unstructured CES service and channel-associated signaling for structured CES, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.

To configure the T1/E1 port for structured CES, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} slot/port`	Enters controller configuration mode for the T1 or E1 controller at the specified *slot/port* location. The prompt changes again to show that you are in controller configuration mode.
Step 2	`Router(config-controller)# clock source {line {primary | secondary} | internal}`	Specifies which end of the circuit provides clocking for the T1 or E1 interface. The keywords are as follows: • **line**—Specifies that the interface will clock its transmitted data from a clock recovered from the line's receive data stream. This is the default. • **primary**—Specifies the source of primary line clocking. The default primary TDM clock source is from the T0 controller. • **secondary**—Specifies the source of secondary line clocking. The default secondary TDM clock source is from the T1 controller. • **internal**—Specifies that the interface will clock its transmitted data from its internal clock. The clock source should be set to use internal clocking when the installed video WAN interface card (VWIC) uses the clocking designated by the CES clock setting.

	Command	Purpose

Step 3	**T1 Line**
	`Router(config-controller)# framing {sf | esf}`
	
	E1 Line
	`Router(config-controller)# framing {crc4 | no-crc4} [australia]`
		Sets the framing for the E1 or T1 data line.
		The keywords are as follows:
		• **sf**—Specifies Super Frame as the T1 frame type.
		• **esf**—Specifies Extended Super Frame as the T1 frame type. This frame type is required for ATM on T1 lines. This setting is automatic for T1 when ATM mode is set.
		• **crc4**—Specifies CRC4 frame as the E1 frame type. This frame type is required for ATM on E1 lines. This setting is automatic for E1 when the ATM mode is set.
		• **no-crc4**—Specifies no CRC4 frame as the E1 frame type.
		• **australia**—(Optional) Specifies the E1 frame type used in Australia.

Step 4	`Router(config-controller)# linecode {ami | b8zs |hdb3}`	Selects the line-code type for T1 or E1 lines. The keywords are as follows: • **ami**—Specifies alternate mark inversion (AMI), which is available for T1 or E1 lines. It represents zeros using a 01 within each bit cell, and ones are represented by 11 or 00, alternately, within each bit cell. AMI requires that the sending device maintain ones density. Ones density is not maintained independently of the data stream. • **b8zs**—Sets the line encoding according to your service provider's instructions. Bipolar-8 zero substitution (B8ZS), available only for T1 lines, encodes a sequence of eight zeros in a unique binary sequence to detect line coding violations. • **hdb3**—Specifies high-density bipolar 3 (HDB3) as the line-code type. It is required for ATM on E1 lines. This setting is automatic for E1 when the ATM mode is set. **Note** When the E1 controller is specified, you must also configure scrambling on the ATM 0 interface. (See Step 3 of the configuration task table in the "Configuring ATM Interfaces to Support Video over PVCs and SVCs" section on page 778.)

	Command	Purpose
Step 5	`Router(config-controller)# ces-clock [adaptive | srts | synchronous]`	Specifies the type of clocking used for T1 interfaces using structured CES. The keywords are as follows: • **adaptive**—Adjusts output clock on a received AAL1 on first-in, first-out basis. Use in unstructured mode. • **srts**—Sets the clocking mode to synchronous residual time stamp. • **synchronous**—Configures the timing recovery to synchronous for structured mode. **Note** Only synchronous clocking can be used with structured CES.
Step 6	`Router(config-controller)# tdm-group tdm-group-no timeslot timeslot-list [type {e&m | fxs [loop-start | ground-start] | fxo [loop-start | ground-start] | fxs-melcas | fxo-melcas | e&m-melcas}`	Configures a list of time slots for creating clear channel groups (pass-through) for time-division multiplexing (TDM) cross-connect. For an explanation of the keywords and arguments, see Step 7 in the "Configuring Video in Pass-Through Mode" configuration task table in this chapter.
Step 7	`Router(config-controller)#exit`	Exits controller configuration mode.
Step 8	`Router(config)#connect connection-name atm slot/port-1 [name of PVC/SVC | vpi/vci] {T1 | E1} slot/port-2 TDM-group-number`	Defines the connections between T1 or E1 controller ports and the ATM interface. The keywords and arguments are as follows: • *connection-name*—Specifies a name for this connection. • **atm**—Specifies the ATM interface. • *slot/port-1*—Specifies the location of the ATM controller to be connected. • *name of PVC/SVC*—Specifies the permanent or switched virtual circuit. • *vpi/vci*—Specifies a virtual path identifier (VPI) and virtual channel identifier (VCI). • **T1**—Specifies a T1 port. • **E1**—Specifies an E1 port.
		• *slot/port-2*—Specifies the location of the T1 or E1 controller to be connected. • *TDM-group-number*—Specifies the number identifier of the time-division multiplexing (TDM) group associated with the T1 or E1 controller port and created by using the **tdm-group** command. Valid values are from 0 to 23 for T1 and from 0 to 30 for E1

Configuring the Proxy and T.120

To configure the Multimedia Conference Manager for voice, video, and data traffic, see the "Configuring H.323 Gatekeepers and Proxies" chapter in this configuration guide.

Note This feature is supported on the Cisco 2600 series, 3600 series, and 7200 series routers and on the Cisco MC3810 multiservice concentrator.

To configure Multimedia Conference Manager for this feature, follow these steps beginning in global configuration mode:

	Command	Purpose/Comment
Step 1	`Router(config)# proxy h323`	Enables the proxy feature on your router.
Step 2	`Router(config)# ip routing`	Makes sure that Fast Switching, which is required for the T.120 feature, is enabled.
Step 3	`Router(config)# interface` *type number* [*name-tag*]	Configures an interface type and enters interface configuration mode. The keywords and arguments are as follows: • *type*—Specifies the type of interface to be configured. (For a list of the interface type keywords, see Table 57, which follows this configuration task table.) • *number*—Specifies the port, connector, or interface card number. On a Cisco 4000 series router, specifies the network process monitor (NPM) number. The numbers are assigned at the factory at the time of installation or when added to a system, and they can be displayed with the **show interfaces** command. • *name-tag*—(Optional) Specifies the logical name to identify the server configuration so that multiple entries of server configuration can be entered. This optional argument is for use with the Redundant Link Manager (RLM) feature.
Step 4	`Router(config-if)# ip route-cache same-interface`	Tells the proxy that when sending the packets out, it should use the same interface that the packets came in on. The packets are sent within the interrupt service context. Otherwise, the packets are queued for processing by the Cisco IOS, which is slower and may lead to packet loss.

	Command	Purpose/Comment
Step 5	`Router(config-if)# h323 interface [port number]`	Selects an interface whose IP address will be used by the proxy to register with the gatekeeper. The *port number* argument specifies the port number on which the proxy will listen for incoming call setup requests. The range is from 1 to 65,356. The default port number for the proxy is 11,720 in -isx- or -jsx- Cisco IOS images. The default port number for the proxy is 1720 in -ix- Cisco IOS images that do not contain the Voice over IP (VoIP) gateway. To use the default port, enter the **no h323 interface** command and then the **h323 interface** command.
Step 6	`Router(config-if)# h323 h323-id`	Specifies the name of the proxy being registered with the gatekeeper. The *h323-id* argument specifies the name of the proxy. It is recommended that this be a fully qualified e-mail identification (ID), with the domain name being the same as that of its gatekeeper. If the proxy has registered successfully on a Cisco gatekeeper, you can see the name of the proxy when you enter the **show gatekeeper endpoints** command.

	Command	Purpose/Comment
Step 7	`Router(config-if)# h323 gatekeeper [id gatekeeper-id] {ipaddr ipaddr [port] | multicast}`	Specifies the gatekeeper associated with a proxy and controls how the gatekeeper is discovered. The keywords and arguments are as follows: • **id** *gatekeeper-id*—(Optional) Specifies the gatekeeper name. Typically, this is a Domain Name System (DNS) name, but it can also be a raw IP address in dotted form. If this parameter is specified, gatekeepers that have either the default or explicit flags set for the proxy's subnet will respond. If this parameter is not specified, only those gatekeepers with the default subnet flag will respond. • **ipaddr** *ipaddr* [*port*]—Specifies that the gatekeeper discovery message will be unicast to this address and, optionally, to the port specified. • **multicast**—Specifies that the gatekeeper discovery message will be multicast to the well-known Registration, Authentication, and Status (RAS) multicast address and port.
Step 8	`Router(config-if)# h323 t120 {bypass | proxy}`	Enables the T.120 capabilities on the router and specifies bypass or proxy mode. The keywords are as follows: • **bypass**—Specifies that the H.245 Open Logical Channel messages for T.120 data channels are passed unmodified through the proxy and that TCP connections for T.120 are established directly between the two endpoints of the H.323 call. • **proxy**—Sets proxy mode. In this mode, T.120 features function properly.

The following table lists the interface types that may be used for the type argument with the interface command.

Table 57 Interface "Type" Keywords

Keyword	Interface Type
async	Port line used as an asynchronous interface.
atm	ATM interface.
bri	ISDN BRI. This interface configuration is propagated to each of the B channels. B channels cannot be individually configured. The interface must be configured with dial-on-demand commands for calls to be placed on that interface.
dialer	Dialer interface.
ethernet	Ethernet IEEE 802.3 interface.

Table 57 Interface "Type" Keywords

Keyword	Interface Type
fastethernet	100-Mbps Ethernet interface on the Cisco 4500, Cisco 4700, Cisco 7000, and Cisco 7500 series routers.
fddi	Fiber Distributed Data Interface (FDDI).
group-async	Master asynchronous interface.
hssi	High-Speed Serial Interface (HSSI).
lex	LAN Extender (LEX) interface.
loopback	Software-only loopback interface that emulates an interface that is always up. It is a virtual interface supported on all platforms. The interface number is the number of the loopback interface that you want to create or configure. There is no limit on the number of loopback interfaces you can create.
null	Null interface.
port-channel	Port channel interface.
pos	Packet OC-3 interface on the Packet over SONET Interface Processor.
serial	Serial interface.
switch	Switch interface.
tokenring	Token Ring interface.
tunnel	Tunnel interface; a virtual interface. The number is the number of the tunnel interface that you want to create or configure. There is no limit on the number of tunnel interfaces you can create.
vg-anylan	100VG-AnyLAN port adapter.

Configuring the Gatekeeper to Support Zone Bandwidth

Gatekeeper support for zone bandwidth is supported on the Cisco 2600 series, 3600 series, and 7200 series routers and on the MC3810 multiservice concentrator.

For more information on configuring gatekeepers to support zone bandwidth, refer to the document *Configuring H.323 VoIP Gatekeeper for Cisco Access Platforms*.

To configure the gatekeeper to support zone bandwidth, use the following commands beginning in gatekeeper configuration mode:

	Command	Purpose/Comment
Step 1	`Router(config)# gatekeeper`	Enters gatekeeper configuration mode.
Step 2	`Router(config-gk)# bandwidth {interzone | total | session} {default | zone zone-name} bandwidth-size`	Specifies the maximum aggregate bandwidth for H.323 traffic. The keywords and arguments are as follows: • **interzone**—Specifies the maximum bandwidth for H.323 traffic between one zone and another zone. • **total**—Specifies the maximum bandwidth for H.323 traffic within a zone and between zones (intrazone and interzone). • **session**—Specifies the maximum bandwidth allowed for a single session in a specific zone or in all zones. • **default**—Specifies the maximum bandwidth for all applicable zones, depending on the keyword with which it is used. • **zone**—Specifies a particular zone. • *zone-name*—Names the particular zone. • *bandwidth-size*—Specifies maximum bandwidth. For interzone and total, the range is from 1 to 10,000,000 kbps. For session, the range is from 1 to 5000 kbps.
Step 3	`Router(config-gk)# bandwidth remote bandwidth-size`	Specifies the total bandwidth for H.323 traffic between this gatekeeper and another gatekeeper. The *bandwidth-size* argument specifies the maximum bandwidth. The range is from 1 to 10,000,000 kbps.

Configuring RSVP-ATM QoS Interworking

RSVP-ATM QoS interworking provides support for controlled load service using RSVP over an ATM core network. This feature requires the ability to signal for establishment of SVCs across the ATM cloud in response to RSVP reservation messages. To meet this requirement, RSVP over ATM supports mapping of RSVP sessions to ATM SVCs. Refer to the document *RSVP-ATM QoS Interworking* for information on how to configure RSVP over an ATM core network.

RSVP-ATM QoS interworking is supported on the Cisco 2600 series, 3600 series, and 7200 series routers and on the MC3810 multiservice concentrator.

Verifying RSVP-ATM QoS Interworking Configuration

Step 1 To see information about the remote bandwidth, enter the **show gatekeeper status** command.

```
Router# show gatekeeper status

Gatekeeper State:UP
Zone Name:        DVM1
Zone Name:        DVM2
Zone Name:        test1
Accounting:       DISABLED
Security:         DISABLED
Maximum Remote Bandwidth:
Current Remote Bandwidth:0 kbps
```

Step 2 To display bandwidth information for all zones, enter the **show gatekeeper zone status** command.

```
Router# show gatekeeper zone status

                          GATEKEEPER ZONES
                          ================
GK name       Domain Name    RAS Address      PORT  FLAGS
-------       -----------    -----------      ----- -----

DVM1          dvm1.com       172.28.129.50    1719  LS
  BANDWIDTH INFORMATION (kbps) :
    Maximum interzone bandwidth :
    Current interzone bandwidth :   0
    Maximum total bandwidth :
    Current total bandwidth :   0
    Maximum session   bandwidth :
  SUBNET ATTRIBUTES :
    All Other Subnets :(Enabled)
  PROXY USAGE CONFIGURATION :
    Inbound Calls from DVM2 :
      to terminals in local zone DVM1 :use proxy
      to gateways in local zone DVM1  :do not use proxy
    Outbound Calls to DVM2 :
      from terminals in local zone DVM1 :use proxy
      from gateways in local zone DVM1  :use proxy
    Inbound Calls from all other zones :
      to terminals in local zone DVM1 :use proxy
      to gateways in local zone DVM1  :do not use proxy
    Outbound Calls to all other zones :
      from terminals in local zone DVM1 :use proxy
      from gateways in local zone DVM1  :do not use proxy
```

```
DVM2         dvm2.com      172.28.129.50   1719  LS
  BANDWIDTH INFORMATION (kbps) :
    Maximum interzone bandwidth :
    Current interzone bandwidth :   0
    Maximum total bandwidth :
    Current total bandwidth :   0
    Maximum session   bandwidth :
  SUBNET ATTRIBUTES :
    All Other Subnets :(Enabled)
  PROXY USAGE CONFIGURATION :
    Inbound Calls from all other zones :
      to terminals in local zone DVM2 :use proxy
      to gateways in local zone DVM2  :do not use proxy
    Outbound Calls to all other zones :
      from terminals in local zone DVM2 :use proxy
      from gateways in local zone DVM2  :do not use proxy

test1        cisco.com     172.28.129.50   1719  LS
  BANDWIDTH INFORMATION (kbps) :   Maximum session   bandwidth :
  SUBNET ATTRIBUTES :
    All Other Subnets :(Enabled)
  PROXY USAGE CONFIGURATION :
    Inbound Calls from all other zones :
      to terminals in local zone test1 :use proxy
      to gateways in local zone test1  :do not use proxy
    Outbound Calls to all other zones :
      from terminals in local zone test1 :use proxy
      from gateways in local zone test1  :do not use proxy

TEST2        test2.com     172.28.129.54   1719  RS
    Maximum interzone bandwidth :
    Current interzone bandwidth :   0
```

Step 3 To display information about the proxy, such as the T.120 mode and what port is being used, enter the **show proxy h323 status** command.

```
Router# show proxy h323 status

        H.323 Proxy Status
        ==================
    H.323 Proxy Feature:Enabled
    Proxy interface = Ethernet0:UP
    Proxy IP address = 172.28.129.50
    Proxy IP port = 11720
    Application Specific Routing:Disabled
    RAS Initialization:Complete
    Proxy aliases configured:
      H323_ID:PROXY
    Proxy aliases assigned by Gatekeeper:
      H323_ID:PROXY
    Gatekeeper multicast discovery:Disabled
    Gatekeeper:
        Gatekeeper ID:DVM1
        IP address:172.28.129.50
    Gatekeeper registration succeeded
    T.120 Mode:PROXY
    RTP Statistics:OFF
    Number of calls in progress:0
```

Video Applications Configuration Examples

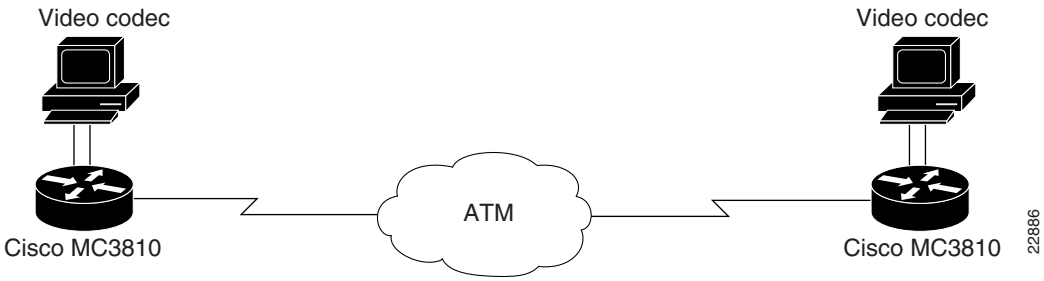
This section provides the following configuration examples:

Video over ATM PVCs and SVCs Configuration Examples

The configuration excerpts in this section illustrate how two Cisco MC3810 multiservice concentrators communicate back-to-back as shown in Figure 139.

These examples focus on the specific requirements of ATM video SVCs and PVCs rather than on the complete ATM setup.

Figure 139 Two Cisco MC3810s Using ATM SVCs or PVCs for Videoconferencing



Initially, the network clocks are set up on each multiservice access concentrator so that video codecs can operate at a multiple of 64 kbps:

Hostname MC3810A

```
!
network-clock base-rate 64k
ip subnet-zero
ip wccp version 2
ip host router 225.255.255.254
!
appletalk routing
ipx routing 1111.0045.0005
```

Hostname MC3810B

```
!
network-clock base-rate 64k
ip subnet-zero
ip wccp version 2
ip host router 225.255.255.254
!
appletalk routing
ipx routing 1111.0045.0002
```

The following commands show the configuration of the T1 0 controller, which is for ATM service. Extended Superframe (ESF) framing and B8ZS are required for ATM. The default clock source is line, and the default for the T1 1 controller automatically becomes internal.

Hostname MC3810A

```
controller T1 0
 framing esf
 linecode b8zs
 mode atm
!
```

Hostname MC3810B

```
controller T1 0
 framing esf
 linecode b8zs
 mode atm
!
```

Serial interface 0 connects to the local video codec. The restart delay is set to 0 minutes so that the hardware is not reset when it goes down. The clock rate of 384 kbps is the speed at which the video images are sent.

Hostname MC3810A

```
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation atm-ces
 no ip route-cache
 no ip mroute-cache
 no keepalive
 serial restart-delay 0
 clockrate network 384000
```

Hostname MC3810B

```
interface Serial0
 no ip address
 no ip directed-broadcast
 encapsulation atm-ces
 no ip route-cache
 no ip mroute-cache
 no keepalive
 serial restart-delay 0
 clockrate network 384000
```

The following commands show how to configure the ATM interface and set up PVCs to supply Q.SAAL signaling and ILMI management for SVC communications. Note that you can also specify the NSAP address by using the **atm video aesa** command with an ESI value.

Hostname MC3810A

```
interface ATM0
 ip address 10.1.1.5 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm ilmi-keepalive
 atm video aesa default
```

Hostname MC3810B

```
interface ATM0
 ip address 10.1.1.6 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm ilmi-keepalive
 atm video aesa default
```

The following examples show dial PVCs for video communications. CBR is required for reliable video. The CBR speed is set at 117 percent of the video data rate of 384 kbps, which is configured on serial interface 0.

Hostname MC3810A

```
pvc 10 32 69
  cbr 449
  encapsulation aal1
```

Hostname MC3810B

```
pvc 11 33 70
  cbr 449
  encapsulation aal1
```

The following examples show dial peers set up for SVC video. Specify local peers through the **port signal** command, which indicates the slot location of the VDM and the port location of the EIA/TIA-366 interface. Enter the **port media** command to specify the serial interface for the codec connection. The two configurations are shown one after the other rather than side by side.

The commands are as follows for MC3810A:

```
dial-peer video 111 videocodec
 nsap 47.0091810000000002F26D4901.00107B4832E1.C8
 port signal 1/0
 port media Serial0
 destination-pattern 121
!
dial-peer video 221 videoatm
 destination-pattern 221
 session target ATM0 svc nsap 47.0091810000000002F26D4901.00107B09C645.C8
```

The commands are as follows for MC3810B:

```
dial-peer video 111 videocodec
 nsap 47.0091810000000002F26D4901.00107B09C645.C8
 port signal 1/0
 port media Serial0
 destination-pattern 221
!
dial-peer video 121 videoatm
 destination-pattern 121
 session target ATM0 svc nsap 47.0091810000000002F26D4901.00107B4832E1.C8
```

CES Video Traffic on the Cisco MC3810 Multiservice Concentrator Configuration Example

The following is an example of configuring video traffic over ATM AAL1 using CES on a Cisco MC3810 multiservice concentrator:

```
network-clock base-rate 64k

controller T1 0
 mode atm

interface Serial0 point-to-point
 no ip address
 encapsulation atm-ces
 clockrate network-clock 768000
 ces connect 25 atm0 pvc 25/100
```

```
interface ATM0 point-to-point
 ip address 223.223.224.229 255.255.255.0
 no ip mroute-cache
 no ip route-cache
 map-group atm1
 pvc 25 25 100
 encapsulation aal1
 cbr 870

no ip classless

map-list atm1
 ip 223.223.224.228 atm-vc 26 broadcast

line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
```

Video Traffic on a Cisco 3600 Series Router Configuration Example

In the following example, the OC-3/STM-1 ATM CES network module is configured for video traffic. This feature is configurable on the Cisco 3600 series routers.

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 3640
!
ces 1/0
 clock-select 1 ATM1/0
!
ip subnet-zero
no ip routing
!
cns event-service server
!
controller T1 1/0
 framing esf
 clock source internal
 linecode b8zs
 cablelength short 133
 tdm-group 0 timeslots 1-6
!
controller T1 1/1
!
interface Ethernet0/0
 ip address 1.2.60.127 255.255.0.0
 ip broadcast-address 1.2.255.255
 no ip route-cache
 no ip mroute-cache
!
interface ATM1/0
 no ip address
 no ip route-cache
 no ip mroute-cache
 no atm ilmi-keepalive
 pvc 0 0/41 ces
!
```

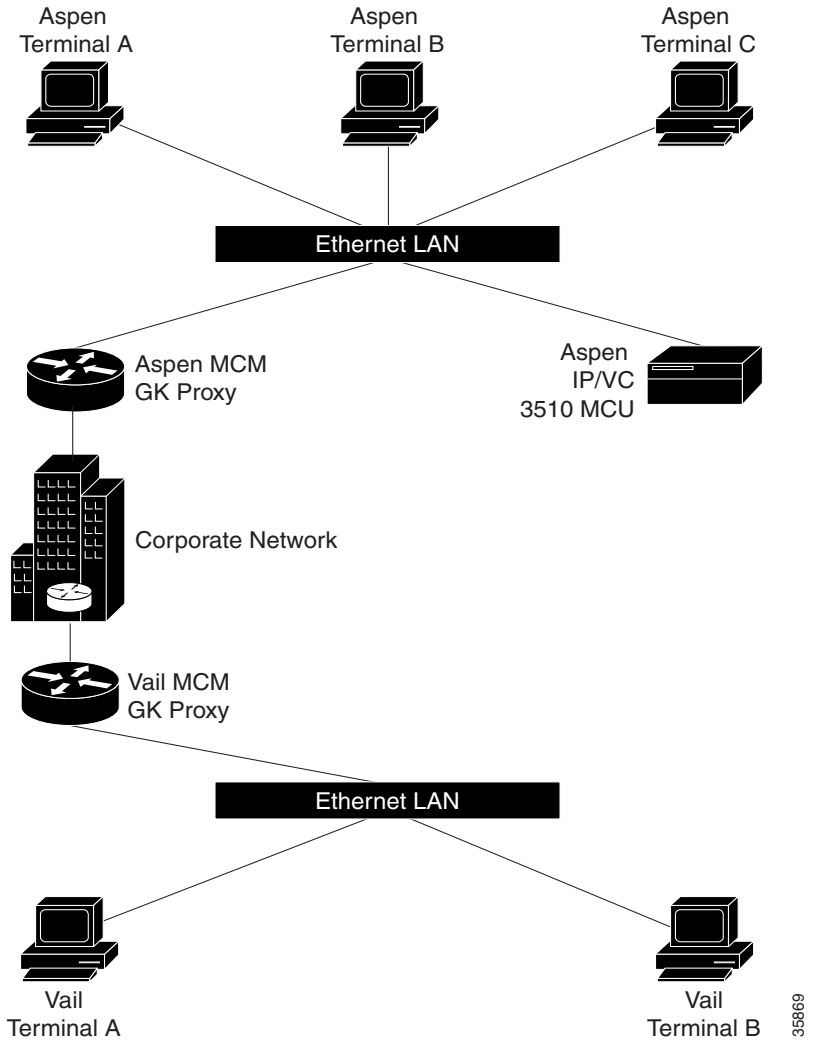
```
ip default-gateway 1.2.0.1
ip classless
ip route 223.255.254.0 255.255.255.0 1.2.0.1
no ip http server
!
connect video-1 ATM1/0 0/41 T1 1/0 0
!
line con 0
 transport input none
line aux 0
line vty 0 4
 login
!
```

Cisco IP/VC 3510 Multipoint Control Unit with Cisco IOS Gatekeeper/Proxy Configuration Example

The Cisco IP/VC 3510 multipoint control unit with Cisco IOS gatekeeper/proxy is configurable on the Cisco 2600 series, 3600 series, and 7200 series routers and on the Cisco MC3810 multiservice concentrator.

The following example shows an interzone calling configuration with two zones defined as Aspen and Vail.

Figure 140 Interzone Calling Configuration with Two Zones



The terminals are H.323 terminals.

The definitions for the above are as follows:

- Aspen Terminal A has an E.164 address of 31.
- Aspen Terminal B has an E.164 address of 32.
- Aspen Terminal C has an E.164 address of 33.

- Aspen IP/VC 3510 multipoint control unit (MCU) has an IP address of 10.0.0.2.
- Aspen IP/VC 3510 MCU has three conference prefixes defined 60, 61, and 62.
- Aspen H.323 Gatekeeper (MCM) Proxy has an IP Address of 10.0.0.1.
- Domain is cisco.com.

Vail Terminal A has an E.164 address of 21. The following is the configuration for Aspen MCM GK Proxy:

```
Hostname         Aspen_MCM_GK_Proxy
Proxy h323
interface Ethernet0/0
ip address 10.0.0.1 255.0.0.0
h323 interface
h323 qos ip-precedence 6
h323 h323-id aspen-proxy
h323 gatekeeper id aspen ipaddr 10.0.0.1
gatekeeper
zone local aspen   cisco.com 10.0.0.1
zone remote vail cisco.com 12.0.0.1
zone prefix aspen 11
zone prefix vail 12
use-proxy aspen default outbound-from gateway
no shutdown
```

The following is the configuration for Vail MCM GK Proxy:

```
Hostname       Vail_MCM_GK_Proxy
Proxy h323
interface Ethernet0/0
ip address 10.0.0.1 255.0.0.0
h323 interface
h323 qos ip-precedence 6
h323 h323-id vail-proxy
h323 gatekeeper id vail ipaddr 12.0.0.1
gatekeeper
zone local vail cisco.com 12.0.0.1
zone remote aspen cisco.com 10.0.0.1
zone prefix aspen 11
zone prefix vail 12
gw-type-prefix 60 hopoff aspen
gw-type-prefix 61 hopoff aspen
gw-type-prefix 62 hopoff aspen
use-proxy aspen default outbound-from gateway
no shutdown
```

In this example, any terminal registered with the Aspen or Vail gatekeeper may participate in a multiparty call with any participant in either zone. For example, Aspen Terminal A could have a conference with Aspen Terminal C and Vail Terminal A by dialing 61555**33**1221. The conference prefix is 61, the conference password is 555, the invite is **, the E.164 address of Aspen Terminal C is 33, the zone prefix to reach the Vail zone is 12, and the E.164 address of Vail Terminal A is 21.

Alternatively, each terminal could independently dial 61555 to join the conference.

CES Clock Configuration Examples

Table 58 shows allowable combinations for CES clocking configuration.

Table 58 CES Clock Configuration Combinations

T1 Controller	ATM Interface	CES Clock	Network Module Status
clock source internal	**no atm clock internal**	**clock-select** 1 ATM x/0	slave to ATM
clock source internal	**atm clock internal**	**clock-select** 2 T1 x/0	slave to T1
clock source internal	**atm clock internal**	**clock-select** 1 Local Oscillator	master clock

The following sample configurations can be used for CES clock settings.

Network Module As Slave to T1 Clock

In this example the OC-3/STM-1 ATM CES network is using the T1 clock.

```
ces 1/0
  clock-select 1 T1 1/0
controller T1 1/0
  clock source internal
interface ATM 1/0
  atm clock internal
```

Network Module As Master Clock

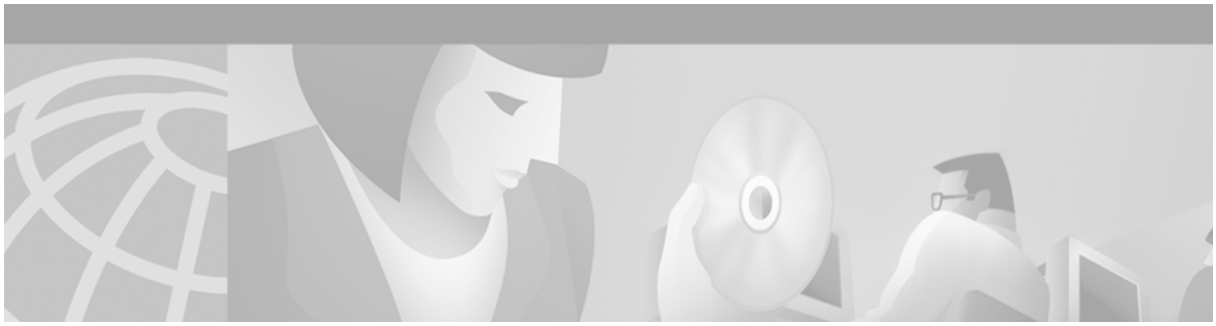
In this example the OC-3/STM-1 ATM CES network module is providing the clock.

```
ces 1/0
  clock-select 1 Local Oscillator
controller T1 1/0
  clock source internal
interface ATM 1/0
  atm clock internal
```

Network Module As Slave to ATM Clock

In this example the OC-3/STM-1 ATM CES network module is using the ATM clock.

```
ces 1/0
  clock-select 1 ATM 1/0
controller T1 1/0
  clock source internal
interface ATM 1/0
  no atm clock internal
```



Configuring Modem Transport Support for VoIP

This chapter explains how to configure modem transport support for Voice over IP (VoIP) and contains the following sections:

For a complete description of the commands used to configure VoIP for modem support, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Note This chapter does not cover modem operation; it covers the transport via VoIP of modem calls and modem call status.

Modem Transport Support Overview

This section describes modem support features. Modem support includes two areas:

Monitoring and Maintaining Modem Call Status

Modem call status is supported by the following features and commands:

- DS-0 busyout traps
- ISDN PRI-requested channel-not-available traps
- Modem health traps

- Show controllers timeslots command
- DS-1 loopback traps

These features allow monitoring and maintaining of access server modem call status at digital signal level zero (DS-0), the PRI bearer channel level, and the modem level.

Modem call status offers the following benefits:

- Improved visibility into the line status of the access server for comprehensive status monitoring and notification capability
- Improved troubleshooting and diagnostics for large dial networks

Note Customers must provide their own management tools.

DS-0 Busyout Traps

A DS-0 busyout trap is generated when any of the following conditions is met:

- A request to busy out a DS-0 occurs
- A busyout completes and the DS-0 is out of service
- A request to take a DS-0 out of busyout mode occurs

DS-0 busyout traps are generated at the DS-0 level for both channel-associated signalling (CAS) and ISDN configured lines.

ISDN PRI-Requested Channel-Not-Available Traps

ISDN PRI-requested channel-not-available traps are generated when a requested DS-0 channel is not available or when there is no modem available to take an incoming call. This feature is available only on ISDN PRI interfaces.

Modem Health Traps

Modem health traps are generated when a modem port is bad, disabled, reflashed, or shut down, or when there is a request to busy out the modem.

show controllers timeslots Command

The **show controllers** command, with the keyword **timeslots,** displays the channel state in detail. This command shows whether the DS-0 channels of a particular controller are in idle, in-service, maintenance, or busyout states. The **show controllers** command applies to both CAS and ISDN PRI interfaces.

DS-1 Loopback Traps

DS-1 loopback traps are generated when a DS-1 line goes into loopback mode.

Modem Pass-Through over VoIP

Modem pass-through over VoIP provides for the transport of modem signals through a packet network by using pulse code modulation (PCM)-encoded packets.

Modem pass-through performs the following functions:

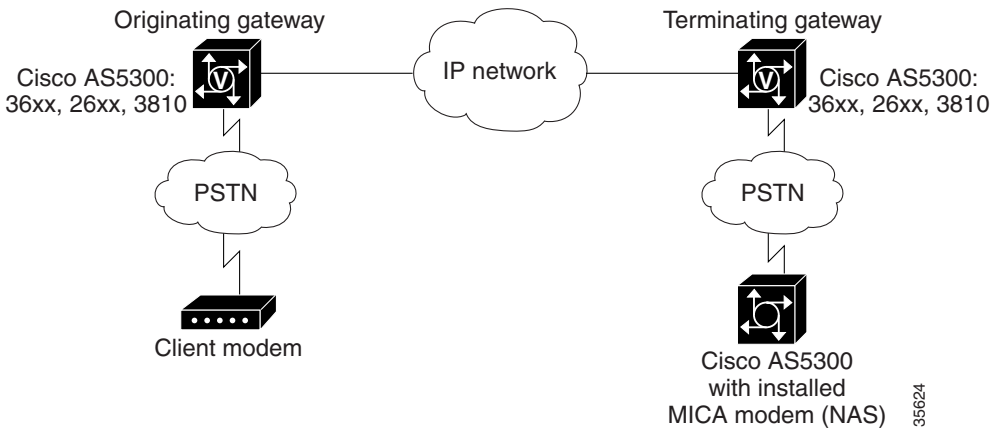
- Repressing processing functions like compression, echo cancellation, high-pass filter, and voice activity detection (VAD)
- Issuing redundant packets to protect against random packet drops
- Providing static jitter buffers of 200 milliseconds (ms) to protect against clock skew
- Differentiating modem signals from voice and fax signals, indicating the detection of the modem signal across the connection, and placing the connection in a state that transports the signal across the network with the least distortion
- Maintaining a modem connection reliably across the packet network for a long duration under normal network conditions

Modem pass-through offers the following benefits:

- Detecting modem tones
- Passing modem signals over the WAN
- Performing proper switchover to pass modem traffic on a bearer channel
- Detecting modems at speeds up to V.90

Figure 141 illustrates the connection from the client modem to a modem ISDN channel aggregation (MICA) technologies modem network access server (NAS).

Figure 141 Modem Pass-Through Connection



Modem Tone Detection

The gateway detects modems operating at speeds up to V.90.

Pass-Through Switchover

See Figure 141. When the gateway detects a data modem, both the originating gateway and the terminating gateway roll over to G.711. The rollover to G.711 disables the high-pass filter, disables echo cancellation, and disables VAD. At the end of the modem call, the voice ports revert to their prior configuration, and the digital signal processor (DSP) goes back to the state it was in before switchover.

For more information about modem pass-through, see the "Configuring Modem Pass-Through" section later in this chapter.

Controlled Redundancy

You can enable payload redundancy so that the modem pass-through over VoIP switchover causes the gateway to emit redundant packets.

Packet Size

When redundancy is enabled, 10-ms sample-sized packets are sent. When redundancy is disabled, 20-ms sample-sized packets are sent.

Clock Slip Buffer Management

When the originating gateway detects a data modem, both the originating gateway and the terminating gateway switch from using dynamic jitter buffers to using static jitter buffers of 200-ms depth. The switch from dynamic to static compensates for Public Switched Telephone Network (PSTN) clocking differences at the originating gateway and the terminating gateway. At the modem call conclusion, the voice ports revert to using dynamic jitter buffers.

Modem Transport Support Prerequisite Tasks

Before configuring your access server to monitor modem call status, perform the following tasks:

- Install the SNMP manager on your workstation.
- Configure the SNMP agent on the access server by entering the following commands:

```
snmp-server community public RO

snmp-server host 10.1.2.3 public
```

For more information on these commands, refer to the *Cisco IOS Configuration Fundamentals Command Reference*.

Before configuring your access server for modem pass-through, perform the following tasks:

- Establish a working VoIP-enabled network.
- Verify network suitability to pass modem traffic. The key characteristics of the network are packet loss, delay, and jitter. These characteristics can be determined by using the Service Assurance Agent (SAA) feature of Cisco IOS software. For more information on SAA, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Modem Transport Support Configuration Task List

To configure modem support, perform the tasks described in the following sections:

Configuring Modem Call Status

To configure modem call status, perform the tasks in the following sections. All four sections are optional.

Enabling DS-0 Busyout Traps

DS-0 busyout traps are supported on the Cisco AS5300 and Cisco AS5800 universal access servers beginning with Cisco IOS Release 12.2. If you are using another Cisco IOS release, use the Feature Navigator on Cisco.com to determine which platforms support this feature.

To generate DS-0 busyout traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps ds0-busyout`	Generates a trap when there is a request to busy out a DS-0 or when busyout finishes. DS-0 busyout traps are disabled by default. The **ds0-busyout** keyword specifies that DS-0 busyout traps be enabled.

Enabling ISDN PRI-Requested Channel-Not-Available Traps

ISDN PRI-requested channel-not-available traps are supported on the Cisco AS5300 and Cisco AS5800 universal access servers beginning with Cisco IOS Release 12.2. If you are using another Cisco IOS release, use the Feature Navigator on Cisco.com to determine which platforms support this feature.

To generate ISDN PRI-requested channel-not-available traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps isdn chan-not-avail`	Generates a trap when the network access server (NAS) rejects an incoming call on an ISDN PRI interface because the channel is not available. ISDN PRI-requested channel-not-available traps are disabled by default. The **isdn chan-not-avail** keywords specify that ISDN PRI-requested channel-not-available traps be enabled.

Enabling Modem Health Traps

Modem health traps are supported on the Cisco AS5300 and Cisco AS5800 universal access servers beginning with Cisco IOS Release 12.2. If you are using another Cisco IOS release, use the Feature Navigator on Cisco.com to determine which platforms support this feature.

To generate modem health traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps modem-health`	Generates a trap when a modem port is bad, disabled, or downloading firmware; when a download fails; when a modem is placed in loopback mode for maintenance; or when there is a request to busy out the modem. Modem health traps are disabled by default. The **modem-health** keyword specifies that modem health traps be enabled.

Enabling DS-1 Loopback Traps

DS-1 loopback traps are supported on the Cisco AS5300 universal access server beginning with Cisco IOS Release 12.2. If you are using another Cisco IOS release, use the Feature Navigator on Cisco.com to determine which platforms support this feature.

To generate DS-1 loopback traps, use the following command in global configuration mode:

Command	Purpose
`Router(config)# snmp-server enable traps ds1-loopback`	Generates a trap when the DS-1 line goes into loopback mode. DS-1 loopback traps are disabled by default. The **ds1-loopback** keyword specifies that DS-1 loopback traps be enabled.

Verifying Enabled Traps

Use the **show running-config** command to verify that the traps are enabled. The following output indicates that all the traps are enabled:

```
.
.
.
Router(config)# show running-config

snmp-server enable traps ds0-busyout
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps modem-health
snmp-server enable traps ds1-loopback
.
.
.
```

Troubleshooting Tips

To troubleshoot the traps, enable debugging for SNMP packets by entering the **debug snmp packets** command in privileged EXEC mode. Check the resulting output to see that the SNMP trap information packet is being sent. The output will vary according to the kind of packet sent or received.

The following example shows the **debug snmp packets** command followed by an excerpt from the debug output. The first and last lines of the sample output show SNMP trap packets that have been sent and received.

```
Router# debug snmp packets

SNMP: Packet received via UDP from 10.5.4.1 on Ethernet0
SNMP: Get-next request, reqid 23584, errstat 0, erridx 0
 sysUpTime = NULL TYPE/VALUE
 system.1 = NULL TYPE/VALUE
 system.6 = NULL TYPE/VALUE
SNMP: Response, reqid 23584, errstat 0, erridx 0
 sysUpTime.0 = 2217027
 system.1.0 = Cisco Internetwork Operating System Software
 system.6.0 =
SNMP: Packet sent via UDP to 10.5.4.1
```

You can also use trap monitoring and logging tools such as **snmptrapd** with debugging flags turned on to monitor output.

Configuring Modem Pass-Through

Modem pass-through over VoIP capability is supported on the Cisco AS5300 universal access server beginning with Cisco IOS Release 12.2. If you are using another Cisco IOS release, use the Feature Navigator on Cisco.com to determine which platforms support this feature.

By default, modem pass-through over VoIP capability and redundancy are disabled.

Tips

For modem pass-through to operate correctly, you must configure modem pass-through in both the originating gateway and the terminating gateway. If you configure only one of the gateways in a pair, the modem call will not be connected successfully.

Redundancy can be enabled in one or both of the gateways. When only a single gateway is configured for redundancy, the other gateway receives the packets correctly but does not produce redundant packets.

Modem pass-through can be configured either globally or for a specific dial peer, or both. If modem pass-through is configured both globally and for a specific dial peer, the dial peer configuration takes precedence over the global configuration. Consequently, when a call matches a particular dial peer, the access server first applies the modem pass-through configuration on the dial peer. Then, if a specific dial peer is not configured, the access server will use the global configuration. The following sections explain further:

- Configuring Modem Pass-Through Globally, page 822
- Configuring Modem Pass-Through for a Specific Dial Peer, page 822

Configuring Modem Pass-Through Globally

To configure modem pass-through for *all* the dial peers of a gateway, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice service voip`	Enters voice-service configuration mode and configures voice service for all gateway connections.
Step 2	`Router(conf-voi-serv)# modem passthrough nse [payload-type number] codec {g711ulaw | g711alaw} [redundancy] [maximum-sessions value]`	Configures modem pass-through for all dial peers of a gateway. The default behavior is **no modem passthrough**. The keywords and arguments are as follows: • **nse**—Used to specify named signalling event (NSE). • **payload-type**—(Optional) The NSE payload type. • *number*—(Optional) Specifies the value of the payload type (from 96 to 119). The default is 100. Use the same payload type for the originating and terminating gateways. When the payload type is 100 and you use the **show running-config** command, the **payload-type** parameter does not appear in the output. • **codec**—Used to specify the type of codec. • **g711ulaw**—Specifies the G.711 u-law codec type. • **g711alaw**—Specifies the G.711 a-law codec type. Use the same codec type for both the originating gateway and the terminating gateway: **g711ulaw** codec is required for T1; **g711alaw** codec is required for E1. • **redundancy**—(Optional) Specifies redundant packets for modem traffic. • **maximum-sessions**—(Optional) Used to specify the maximum number of simultaneous modem pass-through sessions. • *value*—(Optional) Specifies the number of simultaneous modem pass-through sessions (from 1 to 26). The default and recommended value is 16.

Configuring Modem Pass-Through for a Specific Dial Peer

Modem pass-through is disabled by default for all dial peers on the gateway. You can configure modem pass-through on a *specific* dial peer by entering dial-peer configuration mode for the specific dial peer.

You must configure a VoIP dial peer on both the originating and terminating gateways to match the call—for example, using a destination pattern. The modem pass-through parameters associated with those dial peers will then apply to the calls between them.

Note When modem pass-through is configured individually for a specific dial peer, the dial-peer configuration takes precedence over the global configuration for that specific dial peer.

To configure modem pass-through for a specific dial peer, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* **voip**	Enters dial-peer configuration mode and names a specific VoIP dial peer. The keywords and arguments are as follows: • *number*—Selects a particular dial peer. Valid entries are from 1 to 2147483647. • **voip**—Indicates that this is a VoIP peer using voice encapsulation on the plain old telephone service (POTS) network.
Step 2	`Router(config-dial-peer)# modem passthrough {system | nse [payload-type` *number*`] codec {g711ulaw | g711alaw} [redundancy]}`	Configures modem pass-through for a specific dial peer. The default behavior for modem pass-through in dial-peer configuration mode is **modem passthrough system**. **Note** When the **system** keyword is entered, the following parameters are not available: **nse, payload-type, codec,** and **redundancy**. Instead, the values that are used are the ones that were set using the **modem passthrough nse** command in voice-service configuration mode. The keywords and arguments are as follows: • **system**—Causes the gateway to use the values from the global configuration. • **nse**—Used to specify named signalling event. • **payload-type**—(Optional) The NSE payload type. • *number*—(Optional) Specifies the value of the payload type (from 96 to 119). The default is 100. Use the same payload type for the originating and terminating gateways. When the payload type is 100 and you use the **show running-config** command, the **payload-type** parameter does not appear in the output. • **codec**—Used to specify the type of codec. • **g711ulaw**—Specifies the G.711 u-law codec type. • **g711alaw**—Specifies the G.711 a-law codec type. Use the same codec type for the originating and terminating gateways: **g711ulaw** codec is required for T1; **g711alaw** codec is required for E1. • **redundancy**—(Optional) Specifies redundant packets for modem traffic.

Verifying Modem Pass-Through

To verify that modem pass-through is enabled, use the following commands:

- **show running-config** to verify the configuration
- **show dial-peer voice** to verify that modem pass-through over VoIP is enabled

Troubleshooting Tips for Modem Pass-Through

To troubleshoot modem pass-through, perform the following checks:

- Ensure that you can make a voice call.
- Ensure that modem pass-through over VoIP is configured on both the originating gateway and the terminating gateway.
- Ensure that the originating and terminating gateways have the same NSE **payload-type** *number*.
- When two gateways are configured in voice-service configuration mode, ensure that the originating and terminating gateways have the same **maximum-sessions** *value*.
- Use the **debug vtsp dsp** and **debug vtsp session** commands to debug a problem.

Monitoring and Maintaining Modem Pass-Through

To monitor and maintain modem pass-through, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
`Router# show call active voice [brief]`	Displays the voice information for the active call table. The **brief** keyword displays a truncated version.
`Router# show call history voice [brief]`	Displays the voice information for the call history table. The **brief** keyword displays a truncated version.
`Router# show dial-peer voice [number | summary]`	Displays configuration information for dial peers. The keywords and arguments are as follows: • *number*—Specifies a specific dial peer from 1 to 32767. • **summary**—Displays a summary of all dial peers.

Modem Transport Support Configuration Examples

This section provides the following specific configuration examples for modem support:

Modem Call Status Configuration Example

The following example shows sample configuration output with DS-0 busyout traps enabled:

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname router
!
aaa new-model
aaa authentication ppp default group radius
enable password <password>
!
spe 1/0 1/7
 firmware location system:/ucode/mica_port_firmware
spe 2/0 2/7
 firmware location system:/ucode/mica_port_firmware
!
resource-pool disable
!
clock timezone PDT -8
clock calendar-valid
no modem fast-answer
modem country mica usa
modem link-info poll time 60
modem buffer-size 300
ip subnet-zero
!
isdn switch-type primary-5ess
isdn voice-call-failure 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 framing esf
 linecode b8zs
 ds0-group 0 timeslots 1-24 type e&m-fgb
 cas-custom 0
!
interface Loopback0
 ip address 10.5.4.1
!
interface Ethernet0
 no ip address
 shutdown
!
interface Serial0
```

```
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 ip mroute-cache
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no cdp enable
!
interface FastEthernet0
 ip address 10.5.4.1
 duplex full
 speed auto
 no cdp enable
!
interface Group-Async1
 ip unnumbered FastEthernet0
 encapsulation ppp
 ip tcp header-compression passive
 no ip mroute-cache
 async mode interactive
 peer default ip address pool swattest
 no fair-queue
 ppp authentication chap
 ppp multilink
 group-range 1 192
!
interface Dialer1
 ip unnumbered FastEthernet0
 encapsulation ppp
 ip tcp header-compression passive
 dialer-group 1
 peer default ip address pool swattest
 pulse-time 0
 no cdp enable
!
ip local pool swattest 10.5.4.1
ip default-gateway 10.5.4.1
ip classless
!
dialer-list 1 protocol ip permit
snmp-server engineID local 00000009020000D058890CF0
snmp-server community public RO
snmp-server packetsize 2048
snmp-server enable traps pop
snmp-server host 10.5.4.1 public
!
radius-server host 10.5.4.1 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key <password>
!
line con 0
 transport input none
line 1 192
 autoselect ppp
 modem InOut
 transport preferred none
 transport input all
 transport output none
```

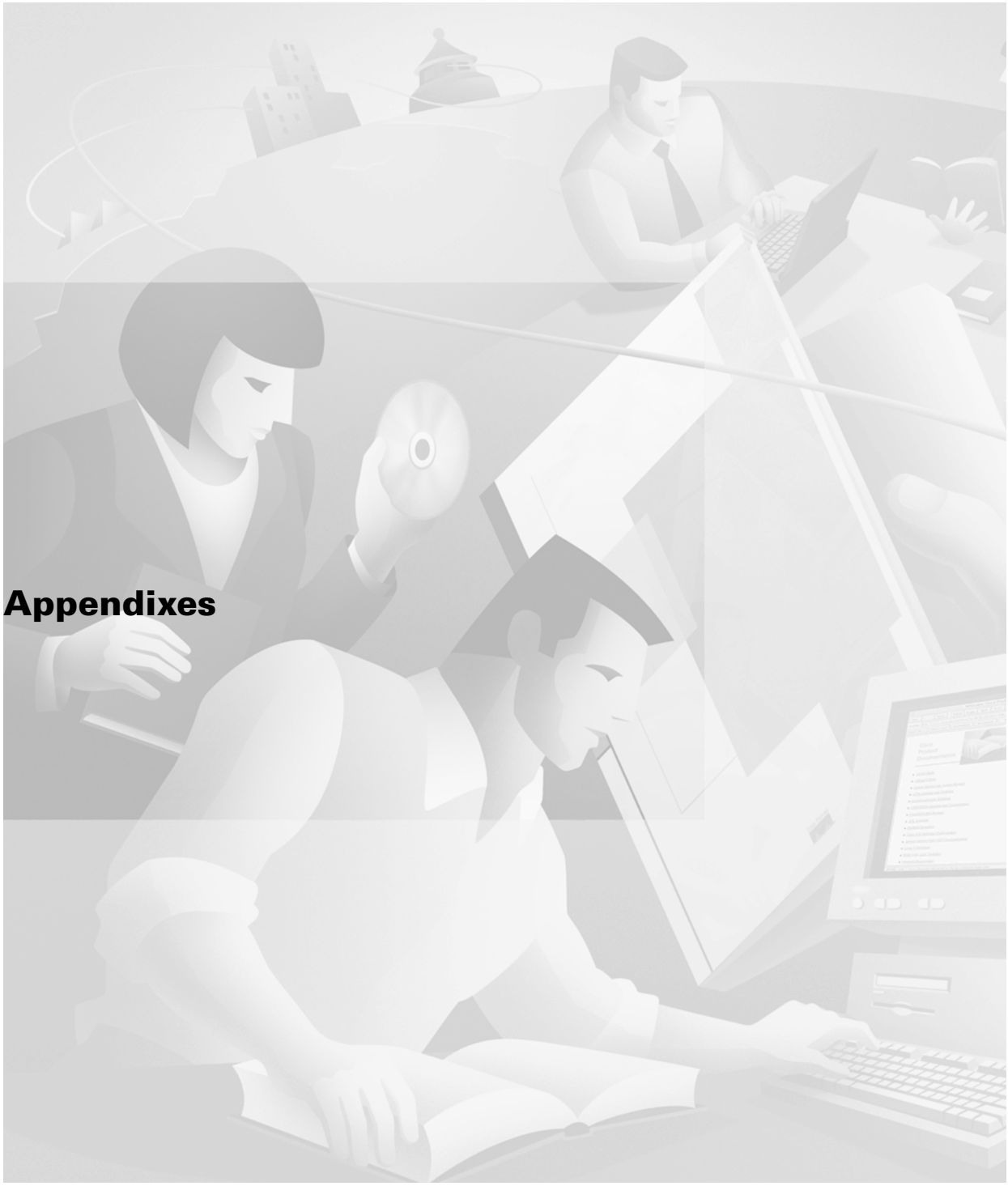
```
line aux 0
line vty 0 4

end
```

Modem Pass-Through Configuration Example

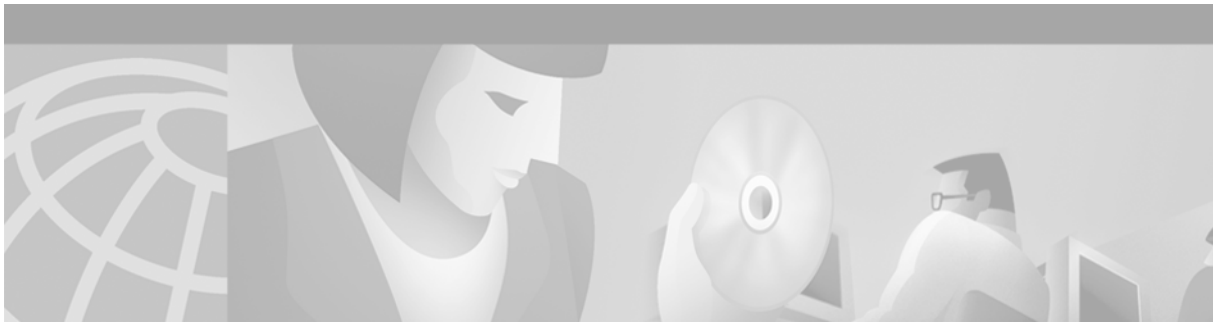
The following example shows a sample configuration for modem pass-through:

```
version 12.2
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
voice service voip
 modem passthrough nse codec g711ulaw redundancy maximum-session 5
!
resource-pool disable
!
ip subnet-zero
ip ftp source-interface Ethernet0
ip ftp username lab
ip ftp password lab
no ip domain-lookup
!
isdn switch-type primary-5ess
cns event-service server
!
mta receive maximum-recipients 0
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 shutdown
 clock source line secondary 1
!
interface Ethernet0
 ip address 1.1.2.2 255.0.0.0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0:23
 no ip address
 encapsulation ppp
 ip mroute-cache
 no logging event link-status
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no peer default ip address
 no fair-queue
 no cdp enable
 no ppp lcp fast-start
!
interface FastEthernet0
 ip address 26.0.0.1 255.0.0.0
 no ip route-cache
 no ip mroute-cache
 load-interval 30
```

```
 duplex full
 speed auto
 no cdp enable
!
ip classless
ip route 17.18.0.0 255.255.0.0 1.1.1.1
no ip http server
!
voice-port 0:D
!
dial-peer voice 1 pots
 incoming called-number 55511..
 destination-pattern 020..
 direct-inward-dial
 port 0:D
 prefix 020
!
dial-peer voice 2 voip
 incoming called-number 020..
 destination-pattern 55511..
 modem passthrough nse codec g711ulaw redundancy
 session target ipv4:26.0.0.2
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```



Appendixes



Configuring Synchronized Clocking

This appendix describes how to configure synchronized clocking. It contains the following sections:

For more information about configuring synchronized clocking, refer to the "Configuring Video Applications" chapter of the *Cisco IOS Voice, Video, and Fax Configuration Guide* and to the *Cisco IOS Wide-Area Networking Configuration Guide*.

For a description of the commands used to configure synchronized clocking, refer to the *Cisco IOS Voice, Video, and Fax Command Reference* and to the *Cisco IOS Wide-Area Networking Command Reference*.

To identify the hardware platform or software image information mentioned in this appendix, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Synchronized Clocking Overview

The Cisco MC3810 multiservice concentrator supports voice and video streams in addition to traditional data streams. Because voice and video streams are real-time streams and originate from synchronous devices, it is important to configure the synchronous clocking to prevent data corruption and data loss.

Due to the real-time nature of voice and video, more configuration and planning is required for voice traffic than is required for traditional data traffic. Because voice and video streams are real-time and continuous, the information is normally generated by the source device and received by the destination device at a synchronized fixed rate. If the source and destination clocking are not synchronized, meaning that the devices generate information at different rates, there will be a loss of information as one side overruns and the other side underruns.

As a result, for voice and video configurations, a single master clock source must be configured to make the network synchronous. The master clock must be used as the clock source for all devices on the network, even when the voice traffic is compressed.

Clocking mismatches can be caused by a variety of configuration problems. The following situations can cause problems:

- Multiple network clock sources that are not synchronized.

  In back-to-back voice systems where the two devices are using different clock sources that are not synchronized, data loss can occur when one device overruns and the other device underruns the voice stream.

  In situations where there is a minor clock mismatch, the Cisco MC3810 may be able to process the mismatch in its internal voice coders in the same way that the voice coders handle minor network delay and jitter. The voice waveform will be degraded but often not noticeably.

  However, when the Cisco MC3810 is using circuit emulation services (CES) to send video traffic, similar clock compensation is not possible because the CES must be in synchronous mode. As a result, when video traffic is sent over a nonsynchronized network, data corruption may occur. This situation will cause video devices connected to the MC3810 to lose frame synchronization and enter a frame-search mode, causing noticeable data loss. Because of these requirements, the network clocks must be synchronized when processing video traffic on the Cisco MC3810.

- Layer 1 conflicts

  Layer 1 conflicts can take place when a Cisco MC3810 with two multiflex trunk modules (MFTs) is placed at the border of two separately clocked T1 or E1 networks and is forced to resolve the clock difference between the networks. As a result, DS1 clock and frame slips can occur, which can result in lengthy reframe times and can cause an attached DS1 device to declare the line down.

Configuring the Cisco MC3810 to a Synchronous Clocked Network

To ensure a synchronized system, you must configure a master clock somewhere within the network and distribute and recover the clock throughout the network. This will allow end devices at opposite ends of the network to reference a common clock source. If you cannot configure a synchronized system, then you can configure multiple clock sources on your network as long as they are accurate enough that the clocking on both clock sources will match.

You can statically configure the Cisco MC3810 to receive or generate clocking using one of the following scenarios:

- Obtain the synchronous clock from a network device attached to controller T1 or E1 0 and distribute the clocking to the other controller and to the universal input/output (UIO) serial ports.
- Obtain the synchronous clock from a network device attached to Controller T1 or E1 1 and distribute the clocking to the other controller and to the UIO serial ports.
- Obtain the synchronous clock from a network device directly attached to serial port 0 (in data terminal equipment [DTE] mode only) and distribute the clocking to the other serial ports and to both controllers.
- Generate the clock internally on the Cisco MC3810 and distribute the clocking to all interfaces.
- When in T1 or E1 mode, all MFTs can provide either line or internal clocking. When one controller is configured to line clocking (obtaining the clocking from the network), the other controller must be configured to internal clocking (obtaining the clocking internally from the other controller).

Note Configuring a clock source from the digital voice module (DVM) is supported if the installed DVM is either hardware version 4.50 or later and the system control board (SCB) is version 6.05 or later. To verify the hardware version of the SCB, enter the **show version** command and check the entry for the Cisco MC3810 processor revision. To verify the hardware version of the DVM, enter the **show controller T1/E1** command and check the HWVersion entry.

For more information on how to configure clocking for these scenarios, see the "Synchronized Clocking Configuration Task List" section later in this appendix.

In addition, you can define a hierarchy of potential clock sources so that when the primary clock source goes down, the Cisco MC3810 can automatically switch to a backup clock source. For more information, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this appendix.

Synchronized Clocking Configuration Task List

Because of the different ways that public switched telephone networks (PSTNs) and data networks provide clocking, there may be incompatibilities when the Cisco MC3810 is used to integrate voice and data networks. As a result, the Cisco MC3810 must synchronize the disparate clocking, and you must be careful in how you configure your clock sources. The clocking can be derived from one of the following sources:

- The PBX
- The video CODEC (for video applications)
- The ATM or Frame Relay WAN carrier
- The Cisco MC3810 internal clock

Depending on the configuration, you must determine how to configure the appropriate interface on the Cisco MC3810 for the clocking configuration. Each interface provides different clocking support, and depending on the interface used, the commands required to configure the clocking are different. You must also determine whether the Cisco MC3810 interface will be the data circuit-terminating equipment (DCE) or the DTE in the configuration.

The following sections provide configuration tasks:

Configuring the Cisco MC3810 to Obtain Clocking from the Network

This section, which describes several scenarios for statically configuring clocking on the Cisco MC3810, includes the following procedures:

- Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to Serial 0, page 841

Note The procedures in this section statically configure the clock source for the interfaces. If the clock source fails, these procedures do not configure a backup clock source. For information on configuring a hierarchy of backup clock sources, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this appendix.

Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to a T1/E1 Controller

When the Cisco MC3810 recovers clocking from a network device attached to a T1 or E1 controller, the clock recovery circuit on the controller will place a recovered 2 MHz clock on the common circuit toward the network clock phased lock loop (PLL). Once the network-clock PLL circuit receives the valid 2 MHz clock from the controller, the network clock PLL synchronizes to the recovered clock and redistributes the clock to the rest of the system. The other T1/E1 controller and the serial ports on the Cisco MC3810 then derive their clocking from the network clock PLL.

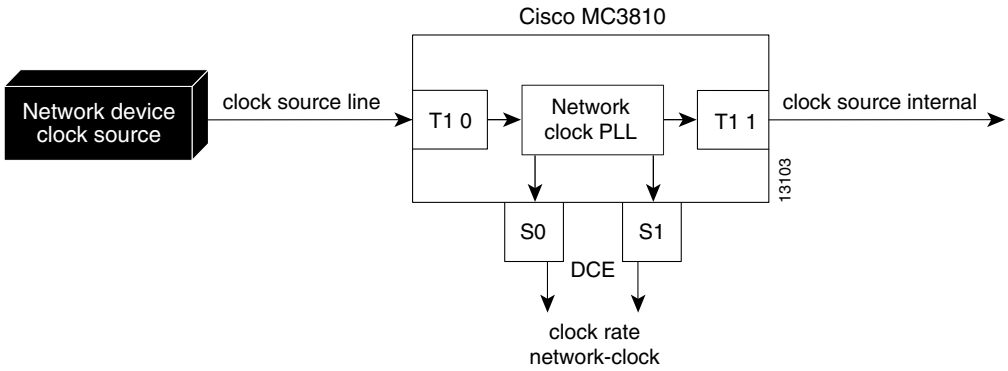
When you configure a T1/E1 controller to recover clocking from a network device, configure the **clock-source** controller configuration command to the **line** setting.

Note Do not configure both T1/E1 controllers to the **line** setting. Doing so will cause both controllers to attempt to drive the network clock PLL at the same time. If you configure both T1/E1 controllers to **line**, there will be clocking conflicts. You will not receive an error message if you incorrectly configure the clocking in this way. Configure one controller for line timing and the other controller for internal or loop timing.

The one exception to this rule is if you configure backup clocks to dynamically activate if the primary clock fails. For more information, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this appendix.

Figure 142 is an example in which the Cisco MC3810 obtains its clock source from a network device attached to controller T1/E1 0 (the MFT).

Figure 142 Obtaining the Clock Source from a Network Device Attached to Controller T1/E1 0



To make sure the network is synchronized, configure the attached network device that obtains its clocking from the Cisco MC3810 (from the T1/E1 controller clock source set to **internal**) to derive its clock from the T1/E1 signal sent by the Cisco MC3810. If the T1/E1 signal received from the attached network device is not synchronous with the Cisco MC3810 network clock, frame and clock slips will occur at the T1/E1 controller, causing loss of data.

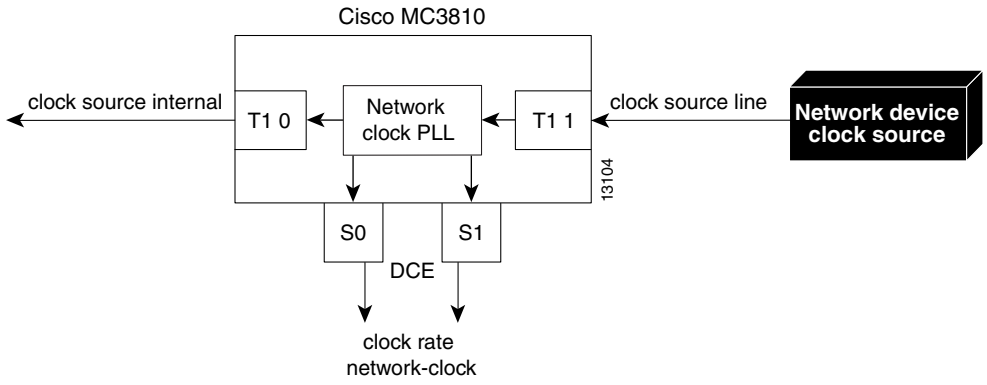
To configure the Cisco MC3810 to obtain its clock source from a network device attached to controller T1/E1 0, use the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for controller T1/E1 0. The *number* argument specifies the network processor module number. The range is 0 through 2.
Step 2	`Router(config-controller)# clock source {line | internal | loop-timed}`	Specifies the clock source of a DS1 link on the Cisco MC3810. (Use the **line** keyword.) Keyword definitions are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed. • **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as data circuit-terminating equipment (DCE) clock sources. This situation assumes that the PBX also takes the clocking from the central office, thereby synchronizing the clocks on the DVM and the MFT.
Step 3	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 4	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for controller T1/E1 1 (see Step 1).
Step 5	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL. (Use the **internal** keyword.) For a full explanation of the keywords, see Step 2.

	Command	Purpose
Step 6	`Router(config-controller)# network-clock base-rate {56k | 64k}`	Configures the network clock base rate for universal I/O serial ports 0 and 1. The keywords are as follows: • **56k**—Sets the network clock base rate to 56 kilobits per second (kbps). • **64k**—Sets the network clock base rate to 64 kbps.
Step 7	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 8	`Router(config)# interface serial number:timeslot`	Enters interface configuration mode for serial 0. The arguments are as follows: • *number*—Specifies the channelized E1 or T1 controller number (0 in the Figure 116 example). • *timeslot*—For ISDN, specifies the D channel time slot, which is :23 channel for channelized T1 and the :15 for channelized E1. PRI time slots are in the range of from 0 to 23 for channelized T1 and in the range of from 0 to 30 for channelized E1. For channel-associated signaling or robbed-bit signaling, specifies the channel group number. The colon (:) is required. On a dual port card, it is possible to run channelized on one port and primary rate on the other port.
Step 9	`Router(config-if)# clock rate network-clock rate`	Configures the network clock speed for serial ports 0 or 1 in DCE mode. The *rate* argument specifies the network clock speed in bits per second. The range is from 56 kbps to 2048 kbps. The value entered should be a multiple of the value set for the **network-clock base-rate** command (see Step 6). (Repeat Steps 8 and 9 for serial port 1.)
Step 10	`Router(config-if)# exit`	Exits interface configuration mode.
Step 11	`Router# show network-clocks`	Displays the network clock configuration.

Figure 143 shows an example in which the Cisco MC3810 obtains its clock source from a network device attached to controller T1/E1 1 (the DVM).

Figure 143 Obtaining the Clock Source from a Network Device Attached to Controller T1/E1 1



To configure the Cisco MC3810 to obtain its clock source from a network device attached to controller T1/E1 1, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for controller T1/E1 1. The *number* argument specifies the network processor module number. The range is 0 through 2.
Step 2	`Router(config-controller)# clock source {line | internal | loop-timed}`	Specifies the clock source of a DS1 link on the Cisco MC3810. (Use the **line** keyword.) Keyword definitions are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed. • **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as data circuit-terminating equipment (DCE) clock sources. This situation assumes that the PBX also takes the clocking from the central office, thereby synchronizing the clocks on the DVM and the MFT.
Step 3	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 4	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1 0 to configure the clock source for the MFT.

	Command	Purpose
Step 5	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 1 to obtain its clocking from the internal network clock phased lock loop (PLL). (Use the **internal** keyword.) For an explanation of the keywords, see Step 2.
Step 6	`Router(config-controller)# network-clock base-rate {56k | 64k}`	Configures the network clock base rate for serial ports 0 and 1. The keywords are as follows: • **56k**—Sets the network clock base rate to 56 kilobits per second (kbps). • **64k**—Sets the network clock base rate to 64 kbps.
Step 7	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 8	`Router(config)# interface serial number:timeslot`	Enters interface configuration mode and specifies the serial 0 interface. The arguments are as follows: • *number*—Specifies the channelized E1 or T1 controller number (0 in the Figure 117 example). • *timeslot*—For ISDN, the D channel time slot, which is :23 channel for channelized T1 and the :15 for channelized E1. PRI time slots are in the range of from 0 to 23 for channelized T1 and in the range of from 0 to 30 for channelized E1.For channel-associated signaling or robbed-bit signaling, the channel group number. The colon (:) is required. On a dual port card, it is possible to run channelized on one port and primary rate on the other port.
Step 9	`Router(config-if)# clock rate network-clock rate`	Configures the network clock speed for serial ports 0 or 1 in DCE mode. The *rate* argument specifies the network clock speed in bits per second. The range is from 56 kbps to 2048 kbps. The value entered should be a multiple of the value set for the **network-clock base-rate** command (see Step 6). (Repeat Steps 8 and 9 for serial port 1.)
Step 10	`Router(config-if)# exit`	Exits interface configuration mode.
Step 11	`Router# show network-clocks`	Displays the network clock configuration.

Configuring a T1/E1 Controller to Loop-Time the Clocking Back to the Network Clock Source

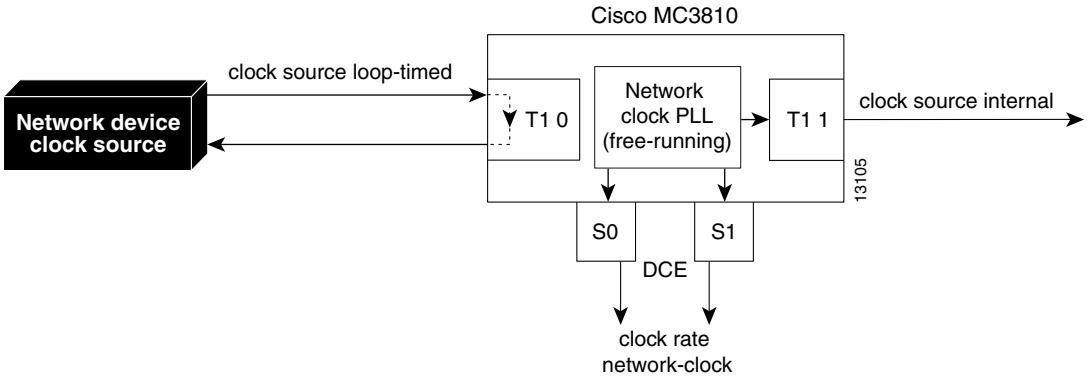
When you configure a T1/E1 controller to loop-time the clocking back to a network device, you configure the **clock-source** controller command to the **loop-timed** setting. The **clock-source** command on the other T1/E1 controller should in most cases be set to the **internal** setting.

When a controller's clock source is set to loop-timed, the internal network clock PLL is placed into free-running mode.

Note Use caution when configuring the controller clock source to loop-timed. This setting should only be used in certain cases, such as when there are two master clocks but you can only obtain clocking from one master clock at a time. Using the functionality to configure a hierarchy of clock sources, you can configure a controller set to loop-timed clock source to become the Cisco MC3810 clock source if the primary clock source fails. For more information about configuring a hierarchy of clock sources, see the "Configuring a Hierarchy of Clock Sources for Backup Purposes" section later in this appendix.

Figure 144 shows an example of a configuration in which the input clock source on the MFT is loop-timed back to the clock source device.

Figure 144 Loop-Timed Clock Source on a T1/E1 Controller



To configure the Cisco MC3810 to use loop-timed clock mode on controller T1/E1 0, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for controller T1/E1 0. The *number* argument specifies the network processor module number. The range is 0 through 2.
Step 2	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 0 to take the clock from the receive line and send it back to the source. (Use the **loop-timed** keyword.) The keywords are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed.

	Command	Purpose
		• **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as data circuit-terminating equipment (DCE) clock sources. This situation assumes that the PBX also takes the clocking from the central office thereby synchronizing the clocks on the DVM and the MFT.
Step 3	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 4	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1 1.
Step 5	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 1 to obtain its clocking from the internal network clock phased lock loop (PLL). For an explanation of the keywords, see Step 2. **Note** To configure controller T1 1 for loop-timed mode, follow the same configuration procedure, but change the controller that will be configured for loop-timed mode.
Step 6	`Router(config-controller)# network-clock base-rate {56k | 64k}`	Configures the network clock base rate for universal I/O serial ports 0 and 1. The keywords are as follows: • **56k**—Sets the network clock base rate to 56 kilobits per second (kbps). • **64k**—Sets the network clock base rate to 64 kbps.
Step 7	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 8	`Router(config)# interface serial number`	Enters interface configuration mode for serial 0.
Step 9	`Router(config-if)# clock rate network-clock rate`	Configures the network clock speed for serial port 0 for DCE mode. The *rate* argument specifies the network clock speed in bits per second. The range is from 56 kbps to 2048 kbps. The value entered should be a multiple of the value set for the **network-clock base-rate** command (see Step 6). Repeat Steps 8 and 9 for serial port 1.
Step 10	`Router(config-if)# exit`	Exits interface configuration mode.
Step 11	`Router# show network-clocks`	Displays the network clock configuration.

Configuring the Cisco MC3810 to Recover Clocking from a Network Device Attached to Serial 0

If serial interface 0 is configured as DTE, it can accept clocking from the attached DCE and use the clocking to drive the network-clock PLL on the Cisco MC3810. The clocking is then distributed to the T1/E1 controllers and to serial interface 1.

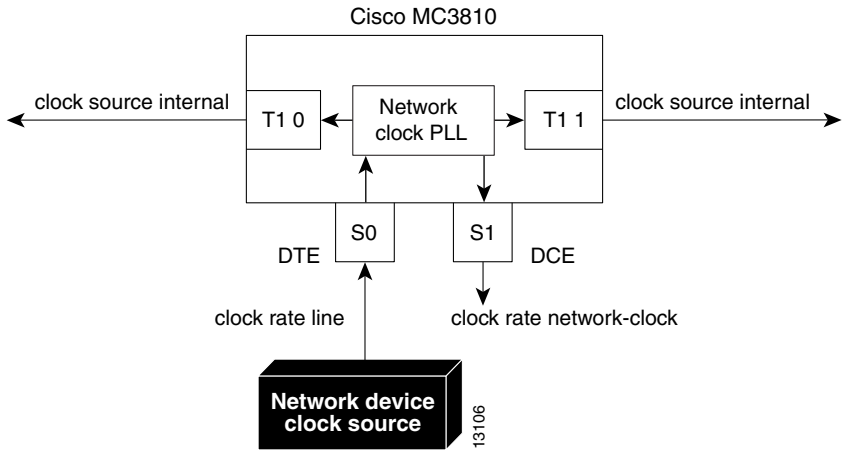
Because the input to the network clock PLL must be 2 MHz, a clock multiplier circuit is used to multiply the incoming clock on serial 0 to 2 MHz in 8 Hz increments. This multiplier is configured using the **clock-rate line** serial interface command. This command is valid only when serial 0 is configured as the DTE device.

Note To recover clocking over serial interfaces, the Cisco MC3810 can recover clocking only from a device attached to serial 0 in DTE mode. It cannot recover clocking from a device attached to serial 1 or to serial 0 in DCE mode.

Note When Q.SIG, ISDN, or the BRI voice module (BVM) is enabled, Serial 1 is normally configured for DCE. If Serial 1 is configured as DTE, you need to make sure that the clock driving serial 1 comes from the same source as the clock driving the system. When Q.SIG, ISDN, or the BVM is enabled, the CPU takes the serial 1 data in time-slot mode that is driven by the system clock. If this clock is different from the clock driving the data into Serial 1, there will be cyclic redundancy check (CRC) errors and the line will not come up.

Figure 145 shows an example of the Cisco MC3810 obtaining clocking from a network device attached to Serial 0.

Figure 145 Clock Source from a Network Device Attached to Serial 0



To configure the Cisco MC3810 to use a network device attached to serial port 0 as the clock source, use the following commands,beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# network-clock base-rate {56k | 64k}`	Configures the network clock base rate for universal I/O serial ports 0 and 1. The keywords are as follows: • **56k**—Sets the network clock base rate to 56 kilobits per second (kbps). • **64k**—Sets the network clock base rate to 64 kbps.
Step 2	`Router(config)# network-clock-select` *priority* `[serial 0 | system | bvm |`*controller*`]`	Configures the network clock PLL to use the multiplied 2 Hz. clock from serial 0. (Set the priority and use the **serial 0** keyword.) The keywords and arguments are as follows: • *priority*—Specifies the priority of the clock source.Valid entries are from 1 to 4. You can configure up to four clock sources. The higher the number of the clock source, the higher the priority. For example, clock source 1 has higher priority than clock source 2. When the higher priority clock source fails, after the delay specified using the network-clock-switch command, the next higher priority clock source is selected. • **serial 0**—(Optional) Specifies serial interface 0 as the clock source. • **system**—(Optional) Specifies the system clock as the clock source. • **bvm**—(Optional) Specifies clocking priority for the BRI voice module. • *controller*—(Optional) Specifies which controller is the clock source. You can specify either the trunk controller (T1/E1 0) or the digital voice module (T1/E1/ 1).
Step 3	`Router(config)# interface serial` *number*	Enters interface configuration mode for serial 0.
Step 4	`Router(config-if)# clock rate line` *rate*	Configures the network clock line rate on serial 0 acting in data terminal equipment (DTE) mode. The rate value is the rate of the incoming clock, and this value must be a multiple of 8 kHz.
Step 5	`Router(config-if)# exit`	Exits interface configuration mode.
Step 6	`Router(config)# interface serial` *number*	Enters interface configuration mode for serial 1.

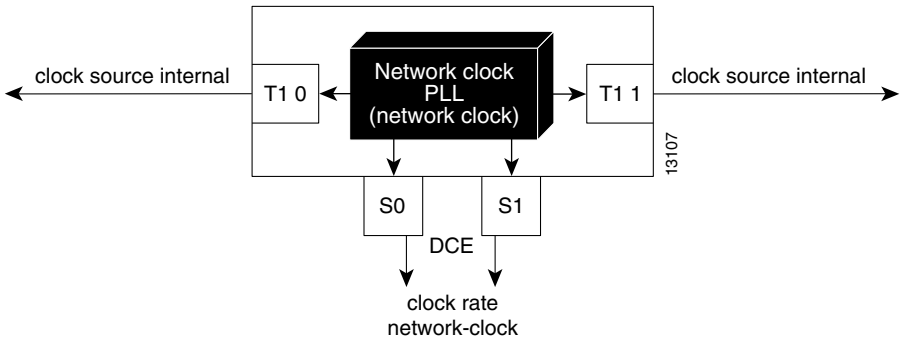
	Command	Purpose
Step 7	`Router(config-if)# clock rate network-clock rate`	Configures the network clock line rate for serial 1 acting in data circuit-terminating equipment (DCE) mode. The rate must be a multiple of the value set with the **network-clock base-rate** command and must match the value set in Step 1.
Step 8	`Router(config-if)# exit`	Exits interface configuration mode.
Step 9	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1 0.
Step 10	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 0 to obtain its clocking from the internal network clock PLL. (Use the **internal** keyword.) The keywords are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed. • **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as DCE clock sources. This situation assumes that the PBX also takes the clocking from the central office, thereby synchronizing the clocks on the DVM and the MFT.
Step 11	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 12	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1 1.
Step 13	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL. (Use the **internal** keyword.) For a full explanation of the keywords, see Step 10 in this configuration task table.
Step 14	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 15	`Router# show network-clocks`	Displays the network clock configuration.

Configuring the Cisco MC3810 to Use the Internal Clock Source

When you configure the Cisco MC3810 to use the internal clock source, the clock source for both T1/E1 controllers is set to **internal** and the master clocking is generated from the Cisco MC3810 2 MHz network clock PLL. The internal clock source is accurate to a Stratum 4 level (plus or minus 0.01 percent).

Figure 146 shows an example of the Cisco MC3810 using its internal clock source and transmitting it outward onto the associated networks.

Figure 146 Using the Cisco MC3810 Internal Clock Source



To configure the Cisco MC3810 to use its internal 2 MHz clock as the clock source, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# network-clock base-rate {56k | 64k}`	Sets the network clock base rate for the serial ports. The default is 56 kilobits per second.
Step 2	`Router(config)# interface serial number`	Enters interface configuration mode for serial 0.
Step 3	`Router(config-if)# clock rate network-clock rate`	Configures the network clock line rate on serial 0 acting in data circuit-terminating (DCE) mode. The rate must be a multiple of the value set with the **network-clock base-rate** command and must match the value set in Step 1.
Step 4	`Router(config-if)# exit`	Exits interface configuration mode.
Step 5	`Router(config)# interface serial 1`	Enters interface configuration mode for serial 1.
Step 6	`Router(config-if)# clock rate network-clock rate`	Configures the network clock line rate on serial 1 acting in DCE mode. The rate must be a multiple of the value set with the **network-clock base-rate** command and must match the value set in Step 1.
Step 7	`Router(config-if)# exit`	Exits interface configuration mode.
Step 8	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1 0.

	Command	Purpose
Step 9	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 0 to obtain its clocking from the internal network clock PLL. (Use the **internal** keyword.) The keywords are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed. • **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as DCE clock sources. This situation assumes that the PBX also takes the clocking from the central office, thereby synchronizing the clocks on the DVM and the MFT.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 11	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1 1.
Step 12	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 1 to obtain its clocking from the internal network clock PLL. (Use the **internal** keyword.) For a full explanation of the keywords, see Step 9.
Step 13	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 14	`Router# show network-clocks`	Displays the network clock configuration.

Note When using the internal Cisco MC3810 clock source as the master clock, make sure to configure any other network devices directly attached to the Cisco MC3810 T1/E1 controllers and serial ports to obtain their clocking from the Cisco MC3810.

Configuring a Hierarchy of Clock Sources for Backup Purposes

The previous configurations apply when a static network clock source is desired with a single clock source. In some conditions, you may want to define a hierarchy of clock sources so that if the primary clock source fails, the system can be configured to use a secondary source rather than to switch to the internal clock (as in the previous configuration sections).

Using the **network-clock-select** command, you can configure a dynamic hierarchy of clock sources that are used if the primary clock source fails. Each clock source is assigned a priority. A higher priority number of a clock source places that source higher in the clocking hierarchy. The highest clock source priority is used as the default.

When a clock source fails, the Cisco MC3810 switches to the clock source in the hierarchy with the next highest priority. For example, if the clock source with priority 1 (the highest priority) fails, the Cisco MC3810 switches to the clock source with priority 2. Then, if the clock source with priority 2 fails, the Cisco MC3810 switches to the clock source with priority 3 (assuming that the clock source with priority 1 has not become active in the meantime.)

If the module providing the clock experiences a failure (for example, if the T1/E1 controller experiences loss of signal or loss of frame), the clock source will be switched.

Note If you shut down a controller that is the current clock source, the shutdown will not cause the clock source to be switched.

To configure a hierarchy of clock sources for backup purposes, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# network-clock-select` *priority* [**serial 0** | **system** | **bvm** | *controller*]	Specifies the highest priority clock source that will provide timing to the system backplane pulse code modulation (PCM) bus. The keywords and arguments are as follows: • *priority*—Specifies the priority of the clock source.Valid entries are from 1 to 4. You can configure up to four clock sources. The higher the number of the clock source, the higher the priority. For example, clock source 1 has higher priority than clock source 2. When the higher priority clock source fails, after the delay specified using the **network-clock-switch** command, the next higher priority clock source is selected. • **serial 0**—(Optional) Specifies serial interface 0 as the clock source. • **system**—(Optional) Specifies the system clock as the clock source. • **bvm**—(Optional) Specifies clocking priority for the BRI voice module. • *controller*—(Optional) Specifies which controller is the clock source. You can specify either the trunk controller (T1/E1 0) or the digital voice module (T1/E1/ 1).

	Command	Purpose
Step 2	`Router(config)# network-clock-switch [switch-delay | never] [restore-delay | never]`	Configure the amount of time the network clock will wait before switching to a different clock and the amount of time the current network clock will wait before recovering. The keywords and arguments are as follows: • *switch-delay*—(Optional) Sets the duration the system waits before switching to the clock source with the next highest priority (as configured with the **network-clock-select** command). • **never**—(Optional) Indicates no delay time before the current network clock source recovers. • *restore-delay*—(Optional) Sets the duration before the current network clock source recovers. • **never**—(Optional) Indicates no delay time before the next priority network clock source is used when the current network clock source fails.
Step 3	`Router(config)# controller {T1 | E1} number`	Enters controller configuration mode for T1/E1. If one of the controllers will be used as a clock source in the hierarchy, enter controller configuration mode for the T1/E1 controller.
Step 4	`Router(config-controller)# clock source {line | internal | loop-timed}`	Configures controller T1/E1 0 to obtain the Cisco MC3810 clock source from an attached network device. (Use the **line** keyword.) Keyword definitions are as follows: • **line**—Specifies that the DS1 link uses the recovered clock. The line value is the default clock source used when the multiflex trunk module (MFT) is installed. • **internal**—Specifies that the DS1 link uses the internal clock. The internal value is the default clock source used when the digital voice module (DVM) is installed. • **loop-timed**—Specifies that the T1/E1 controller will take the clock from the Rx (line) and use it for Tx. This setting decouples the controller clock from the system-wide clock set with the **network-clock-select** command. The loop-timed clock enables the DVM to connect to a PBX and to connect the MFT to a central office when both the PBX and the central office function as data circuit-terminating equipment (DCE) clock sources. This situation assumes that the PBX also takes the clocking from the central office, thereby synchronizing the clocks on the DVM and the MFT.

	Command	Purpose
		If the other controller will be used as a potential clock source in the hierarchy, repeat Steps 3 and 4 in this configuration task table. **Note** To prevent clock source conflicts, make sure to configure both controllers to **clock source line** *after* configuring the **network-clock-select** commands. For more information about how clock source conflicts are resolved using this feature, see the section following this procedure.
Step 5	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 6	`Router(config)# interface serial` *number*	Enters interface configuration mode. If serial interface 0 will be used as a potential clock source in the hierarchy, enter interface configuration mode for serial 0.
Step 7	`Router(config-if)# clock rate line` *rate*	Configures the network clock line rate on serial 0 acting in data terminal equipment (DTE) mode. The rate must be a multiple of the value set with the **network-clock base-rate** command.
Step 8	`Router(config)# exit`	Exits interface configuration mode.
Step 9	`Router# show network-clocks`	Displays the network clock configuration.

When you configure a hierarchy of clock sources, each potential clock source must be preconfigured to a mode that enables the Cisco MC3810 to derive the clock from that source. For example, if a controller will be a potential clock source, the controller clock source must be configured to **line**. If the controller clock source is configured to **internal**, the controller cannot be configured as a potential backup clock source using the **network-clock-select** command.

In the normal configuration, configuring both controllers to clock source line causes clocking conflicts. However, when configuring a hierarchy of clock sources, because only one controller is used as the primary clock source at one time, the conflict is prevented.

The following rules apply to configuring the clock source hierarchy:

- If a controller is a potential clock source in the hierarchy, the controller clock source must be configured to **line**.
- If a controller is a potential clock source in the hierarchy but is not currently being used as the clock source, the clock source setting for that controller is automatically switched to **loop-timed**. This is a temporary state set by the software to prevent a clocking conflict. If the controller becomes the clock source because another clock source fails, the clock source setting for the controller switches to **line**.

  In this situation, even though the setting for the controller clock is switched to loop-timed, the actual configuration remains **line**. This is the difference between the preconfigured state and the temporary "set state" of the controller.
- If either controller is the active clock source, the network clock PLL switch is thrown in the direction of the active clock. The system clock is recovered from the controller with the active clock source.

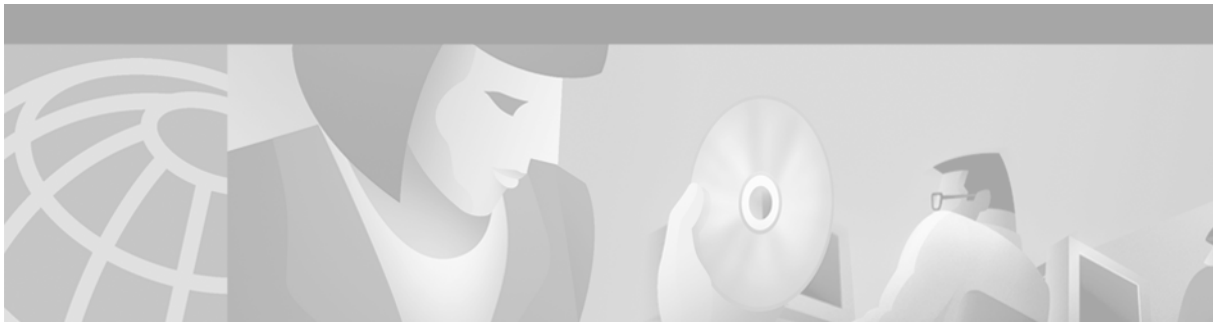
- If serial interface 0 is the active clock source, the clock source settings for both controllers are automatically set to **loop-timed** and the network clock PLL switch is thrown in the direction of the serial port. The system clock is driven by a clock recovered from the DTE serial 0 interface, which has been multiplied from (n x 8000) Hz to 2 MHz.
- If the internal system clock is the active clock source, the clock source settings for both controllers are automatically set to **loop-timed,** and the network clock PLL switch is thrown in the direction of the controllers. Because both controllers are in the **loop-timed** state, neither clock provides a recovered clock to drive the PLL, resulting in a free-running, or internally timed, system clock.

The following is a configuration example showing a hierarchy of clock sources:

```
network-clock-select 1 t1 0
network-clock-select 2 t1 1
network-clock-select 3 serial0
network-clock-select 4 system
network-clock-switch 10 10

controller t1 0
 clock source line

controller t1 1
 clock source line

interface serial0
 clock rate line 64000
```

In this configuration, controller T1 0 is the primary clock source, and the clock source is configured to **line**. Controller T1 1 is a backup clock source and although the clock source is configured to **line**, the system temporarily sets the clock source to the **loop-timed** state.

If the controller T1 0 clock source fails, the system switches to use controller T1 1 as the clock source. The clock source **loop-timed** “set state” on controller T1 1 is switched to the preconfigured **line** state.



Caller ID on Cisco 2600 and 3600 Series Routers and Cisco MC3810 Multiservice Concentrators

This appendix describes Cisco IOS configuration for caller ID as supported on the Cisco MC3810 multiservice concentrator and on Cisco 2600 and 3600 series routers. It includes the following sections:

To identify the hardware platform or software image information associated with a feature in this appendix, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the “Ideentifying Supported Platforms” section in the “Using Cisco IOS Software” chapter.

Called ID Overview

Caller ID (sometimes called *CLID* or *ICLID* for incoming call line identification) is an analog service offered by a central office (CO), which supplies calling party information to subscribers. Typically, the calling party number, and sometimes the name, appears on a station (also called *extension*) device such as a PC telephony software application screen or the display on a telephone. Type 1 caller ID provides the calling party information while the call is ringing, and Type 2 caller ID provides the additional convenience of calling number display while the recipient is on another call. In this release, Cisco provides only Type 1 caller ID support.

The caller ID feature supports the sending of calling party information from foreign exchange station (FXS) loop-start and ground-start ports into a caller ID equipped telephone device. The FXS port emulates the extension interface of a private-branch exchange (PBX) or the subscriber interface for a CO switch.

The caller ID feature supports receiving calling-party information at foreign exchange office (FXO) loop-start and ground-start ports. The FXO port emulates a connection to a telephone and allows connection to a PBX extension interface or (where regulations permit) a CO subscriber line.

The following are benefits of using caller ID:

- Enterprises—Caller ID is invaluable for increasing efficiency through its use in computer telephony integration (CTI) applications, where for example, calling party information can be used to retrieve client information from a database when a customer call is received.

- Service Provider—In traditional telephony, caller ID is a standard service that service provider customers expect. With the Cisco support for caller ID, service providers can offer the feature for packet-switched Voice over IP (VoIP), Voice over Frame Relay (VoFR), and Voice over ATM (VoATM) services.

Calling Name and Number

Figure 147 shows a hypothetical topology where users, indicated by telephone icons, receive different types of caller-ID support depending upon whether the caller-ID information from the caller passes through an FXO or FXS port before reaching the party who receives the call.

Figure 147 Caller ID and ANI Support

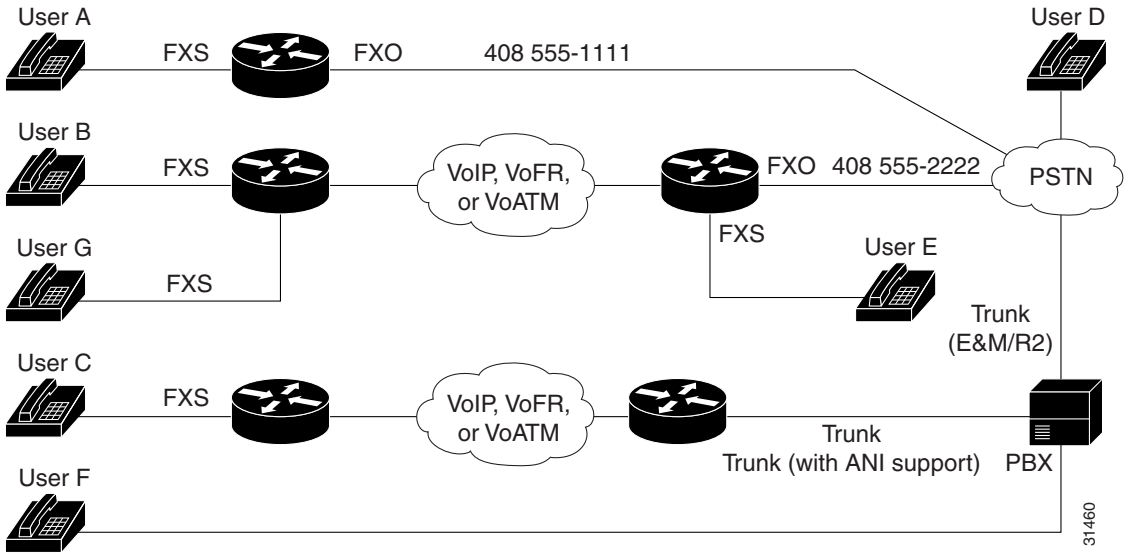


Table 59 shows how caller-ID information is received by the users in Figure 147 as follows:

- When an outbound caller-ID call is carried by a trunk with E&M or R2 signaling, the recipient sees only the ANI calling number of the caller.
- When caller-ID information is sent from an originating FXS station by way of the PSTN, the recipient sees only the identification of the FXO port through which the call is routed.
- When caller-ID information is sent from an originating station over a trunk with E&M or R2 signaling and through the PSTN, the recipient sees only the trunk identification because the ANI information is not preserved by the PSTN.

Table 59 Caller-ID Information Received

User Originating Call	User Receiving Call	Caller-ID Information Received
A	D	D receives the caller ID of the PSTN subscriber line only (408 555-1111).
D	A	A receives the calling number and name of D, provided that the PSTN subscriber line (408 555-1111) is enabled for Caller ID.
D	B	B receives the calling number and name of D, provided that the PSTN subscriber line (408 555-1111) is enabled for Caller ID.
B	D	D receives the caller ID of the PSTN subscriber line only (408 555-2222).
B	E	E receives the Calling Number and Name string of B.
B	G	G receives the Calling Number and Name string of B.
E	B	B receives the Calling Number and Name string of E.
F	C	C receives the Calling Number of F.
C	F	Calling Number of C.
D	C	C Receives Calling Number of D through ANI.
C	D	Calling Number of C goes through ANI to the PSTN. However, the PSTN displays only the trunk ID, so D sees only this information.
C	F	The information that F receives depends on the PBX features available.

Call Time Display

When caller-ID information is sent, the local time set on the router is transmitted with the station name and number. If a call received on an FXO port is terminated on an FXS port, the calling time received on the FXO port is replaced by the local time while transmitting caller ID to the FXS port. This is also true for calls received from the network. The router should be configured to retrieve network time at boot up from an NTP server in order to maintain the correct local time setting.

For more information about voice configuration, refer to the following:

- *Cisco IOS IP Routing Configuration Guide*
- *Cisco IOS Wide-Area Networking Configuration Guide*

The following online feature documentation and installation guides describe the configuration and installation of hardware components:

- For information about installing Cisco MC3810 multiservice concentrators, see *Cisco MC3810 Multiservice Concentrator Hardware Installation* at the following location: http://www.cisco.com/univercd/cc/td/doc/product/access/multicon/3810hwig/index.htm
- For information about installing Cisco 2600 series routers, see the documents listed at the following location: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/index.htm
- For information about installing Cisco 3600 series routers, see the documents listed at the following location: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/index.htm

Caller ID Prerequisites Tasks

The following hardware, software, and basic configurations are required to support caller ID:

- Cisco IOS Release 12.1(3)T software.
- Caller ID service from your service provider.
- A working network. For more information, see the following publications:
  - Cisco IOS *Cisco IOS IP Routing Configuration Guide*
  - Cisco IOS *Wide-Area Networking Configuration Guide*
- Your company's dial plan.
- A working telephony network based on your company's dial plan:
- If applicable to your network, install a 2-channel analog plain old telephone service (POTS) FXS voice interface card (VIC) in a Cisco 2600 series chassis slot or Cisco 2600 or 3600 network module.
- If applicable to your network, install one of the following Cisco MC3810 multiservice concentrator FXO network modules:
  - MC3810-APM-FXO (generic); caller ID is supported in versions v04.xx and later of this APM.
  - MC3810-FXO-PR2 (Pacific Rim 2); caller ID is supported in versions v02.xx and later of this APM.
  - MC3810-FXO-PR3 (Pacific Rim 3); caller ID is supported in versions v02.xx and later of this APM.
  - MC3810-FXO-UK (UK); caller ID is supported in versions v03.xx and later of this APM.
  - MC3810-FXO-GER (Germany); caller ID is supported in versions v03.xx and later of this APM.
- For a Cisco MC3810 multiservice concentrator, install an HCM as follows:
  - An HCM2 to supply 4 or 8 voice or fax channels at high or medium codec complexity.
  - An HCM6 to supply 12 or 24 voice or fax channels at high or medium codec complexity.
- For information about installing Cisco MC3810 multiservice concentrator HCMs, refer to *Cisco MC3810 Multiservice Concentrator Hardware Installation* at the following url: http://www.cisco.com/univercd/cc/td/doc/product/access/multicon/3810hwig/index.htm

  Note The Cisco MC3810 multiservice concentrator voice-compression module does not support caller ID. Install an HCM instead.

- One other network module or WAN interface card to provide the connection to the LAN or WAN.

Caller ID Configuration Task List

Voice-port configuration is the only special configuration required to support caller ID. To configure your voice network fully, use the reference information in the section "Caller ID Prerequisites Tasks" section on page 854 to perform the following tasks on your routers:

- Configure your IP, ATM, or Frame Relay network to support real-time voice traffic.
- Configure voice cards for codec settings.
- Configure voice dial peers. Each dial peer defines the characteristics associated with a call leg.

The remainder of this section describes the steps required to configure caller ID on FXS and FXO voice ports.

Configuring Voice Ports to Support Caller ID

To configure voice ports to support caller ID, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice-port slot/port`	Enters voice-port configuration mode on a Cisco MC3810 multiservice concentrator. The *slot* number for analog voice ports on the Cisco MC3810 multiservice concentrator is always 1. There is no port 0 for voice ports.
	or	
	`Router(config)# voice-port slot-number/subunit-number/port`	Enters voice-port configuration mode on a Cisco 2600 or 3600 series router.
Step 2	`Router(config-voiceport)# connection {plar | tie-line | plar-opx} digits {trunk digits [answer-mode]}`	Specifies the voice-port connection type and the destination telephone number. The keywords and arguments are as follows: • **plar**—Specifies private line automatic ringdown. • **tie-line**—Specifies a tie-line connection to a PBX. • **plar-opx**—Specifies a PLAR off-premises extension (the local voice port provides a local response before the remote voice port receives an answer). • **trunk**—Specifies a straight tie-line connection to a PBX. • **answer-mode**—Indicates whether a trunk connection is specified. The router should not attempt to initiate a trunk connection, but should wait for an incoming call before establishing the trunk. • *digits*—Specifies the destination telephone number.

	Command	Purpose
Step 3	`Router(config-voiceport)# voice confirmation-tone`	Enables the two-beep confirmation tone that a caller hears when picking up the handset, if **connection plar** or **connection plar-opx** is configured.
Step 4	`Router(config-voiceport)# dial-type {dtmf | pulse}`	(For FXO ports only) Selects the appropriate dial type for out-dialing.
Step 5	`Router(config-voiceport)# signal {loop-start | ground-start}`	Selects the appropriate signal type for this interface.
Step 6	`Router(config-voiceport)# codec {g729r8 | g729ar8 | g726r32 | g711alaw | g711ulaw}`	(Cisco MC3810 multiservice concentrator only) Configures the voice-port compression mode. The **g729ar8** value is the default and is recommended. **Note** On Cisco 2600 and 3600 series routers, codec configuration is performed on dial peers. On all supported routers, **codec** command options may vary depending upon the voice card settings.
Step 7	`Router(config-voiceport)# cptone` *locale*	Selects the appropriate voice call progress tone for this interface. Caller ID requires this setting. The caller ID standard (Bellcore/Telcordia, ETSI, ETSI-DTMF) is determined by this command. On the Cisco MC3810 multiservice concentrator, the default setting for *locale* is northamerica. On Cisco 2600 and 3600 series routers, the default setting for *locale* is us. See Table 60 on page 856 for a list of options.
Step 8	`Router(config-voiceport)# ring frequency {25 | 50}`	(Required on Cisco 2600 and 3600 series routers FXS ports only) Selects the appropriate ring frequency (in hertz) specific to the equipment attached to this voice port.
Step 9	`Router(config-voiceport)# caller-id attenuation attenuation`	(Optional on FXO ports only) Specifies an attenuation other than the default of 14 dB (minus 14 dBm), enter a value of from 0 to 64, in decibels.
Step 10	`Router(config-voiceport)# ring number` *number*	(Required on Cisco 2600 and 3600 series routers FX0 ports only) Specifies the maximum number of rings to be detected before answering a call.

The following table lists the options that may be used for the *locale* variable with the **cptone** command.

Table 60 cptone Command Entries for the Cisco 2600 and 3600 Series

Command Option	Country	Command Option	Country
ar	Argentina	lu	Luxembourg
au	Australia	my	Malaysia
at	Austria	mx	Mexico
be	Belgium	nl	Netherlands

Table 60 cptone Command Entries for the Cisco 2600 and 3600 Series (continued)

Command Option	Country	Command Option	Country
br	Brazil	nz	New Zealand
ca	Canada	no	Norway
cn	China	pe	Peru
co	Colombia	ph	Philippines
cz	Czech Republic	pl	Poland
dk	Denmark	pt	Portugal
fi	Finland	ru	Russian Federation
fr	France	sg	Singapore
de	Germany	sk	Slovakia
gr	Greece	si	Slovenia
hk	Hong Kong	za	South Africa
hu	Hungary	es	Spain
is	Iceland	se	Sweden
in	India	ch	Switzerland
id	Indonesia	tw	Taiwan
ie	Ireland	th	Thailand
il	Israel	tr	Turkey
it	Italy	gb	Great Britain
jp	Japan	us	United States
kr	Korea Republic	ve	Venezuela

Configuring FXS and FXO Voice Ports to Support Caller ID

To configure caller-ID on FXS and FXO voice ports, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# caller-id enable`	Enables caller ID. This command applies to FXS voice ports that send caller-ID information and to FXO ports that receive it. By default caller ID is disabled. **Note** If the **station name**, **station number**, or a **caller-id alerting** command is configured on the voice port, these automatically enable caller ID, and the **caller-id enable** command is not necessary.
Step 2	`Router(config-voiceport)# station name` *name*	Configures the station name on FXS voice ports connected to user telephone sets. This sets the caller-ID information for on-net calls originated by the FXS port. You can also configure the station name on an FXO port of a router for which incoming Caller ID from the PSTN subscriber line is expected. In this case, if no caller-ID information is included on the incoming PSTN call, the call recipient receives the information configured on the FXO port instead. If the PSTN subscriber line does provide caller-ID information, this information is used and the configured station name is ignored. The *name* argument is a character string of 1 to 15 characters identifying the station. **Note** This command applies only to caller-ID calls, not Automatic Number Identification (ANI) calls. ANI supplies calling number identification only.

	Command	Purpose
Step 3	`Router(config-voiceport)# station number number`	Configure the station number on FXS voice ports connected to user telephone sets. This sets the caller-ID information for on-net calls originated by the FXS port. You can also configure the station number on an FXO port of a router for which incoming caller ID from the PSTN subscriber line is expected. In this case, if no caller-ID information is included on the incoming PSTN call, the call recipient receives the information configured on the FXO port instead. If the PSTN subscriber line does provide caller-ID information, this information is used and the configured station name is ignored. If the caller-ID station number is not provided by either the incoming PSTN caller ID or by the station number configuration, the calling number included with the on-net routed call is determined by Cisco IOS software by using a reverse dial-peer search. In this case, the number is obtained by searching for a POTS dial-peer that refers to the voice-port and the destination-pattern number from that dial-peer is used. *number* is a string of 1 to 15 characters identifying the station telephone or extension number.
Step 4	`Router(config-voiceport)# caller-id block`	(FXS ports only) When this command is configured at the originating end of a call, it requests that the originating calling party information not be displayed at the called party's telephone. **Note** The calling party information is included in the routed on-net call, as this is often required for other purposes, such as billing and call blocking. The request to block display of the calling party information on terminating FXS ports will normally be accepted by Cisco routers, but no guarantee can be made regarding the treatment by other equipment. This command affects all calls sent to an FXO station from the configured FXS station. The central office (CO) may supply a feature code that a user can dial in order to block caller-ID transmission on a call-by-call basis. When a blocked-information call passes through an FXO interface on the way to its destination, the blocking is passed on to the receiving party.

To configure the alerting method, use the following commands beginning in global configuration mode. Configuration of the alerting method is required when the caller ID standard, specified by locale through the **cptone** command, is other than Bellcore/Telcordia (if you do not configure the alerting method, the default c**aller-id alerting ring 1** command is applied). The command that you enter is determined by the Bellcore/Telcordia or ETSI standard that your service provider uses for caller ID. For more information about standards, see the Caller ID Prerequisites Tasks, page 854 section.

	Command	Purpose
Step 1	`Router(config)# voice-port` *slot/port* or `Router(config)# voice-port` *slot-number/subunit-number/port*	Enters voice-port configuration mode on a Cisco MC3810 multiservice concentrator. The slot number for analog voice ports on the Cisco MC3810 multiservice concentrator is always 1. There is no port 0 for voice ports. Enters voice-port configuration mode on a Cisco 2600 or 3600 series router.
Step 2	`Router(config-voiceport)# caller-id alerting ring {1 | 2}`	Configure this command on FXO ports where caller ID information is received from a subscriber telephone line. and on FXS voice ports from which caller ID information is transmitted to an attached telephone device. Compatible settings are required on both ends of the telephone line connection or caller ID information may not be displayed. Enter **1** if your telephone line service provider or telephone device specifies it, to provide or expect caller ID information following the first ring at the receiving station. This is the default setting. Enter **2** to provide or expect caller-ID information during the long ring pause following two short rings. This setting is used in Australia and the United Kingdom.
Step 3	`Router(config-voiceport)# caller-id alerting line-reversal`	(FXS ports only) Configure this setting only when the attached telephone device requires line polarity reversal to signal the start of caller-ID information transmission.

	Command	Purpose
Step 4	`Router(config-voiceport)# caller-id alerting dsp-pre-alloc`	(FXO ports, only when caller-ID alerting line-reversal is required) Configure this command on the FXO port when the incoming subscriber telephone line uses line polarity reversal to signal the start of caller-ID information transmission. The Cisco FXO interface cannot detect line-reversal alerting in the on-hook state. For this reason, DSPs must be pre-allocated to serve the Type 1 caller ID information when it arrives. Preallocating the DSPs enables the DSP to continuously monitor for the arrival of caller-ID information.
Step 5	`Router(config-voiceport)# caller-id alerting pre-ring`	(FXS ports only) Configure this setting only when the attached telephone device requires the pre-ring (immediate ring) method to signal the start of a caller ID information. The command activates a 250-ms pre-ring.

Verifying Caller ID on Voice Ports Configuration

To verify voice-port configuration, enter the **show voice-port** command. You can specify a voice port or view the status of all configured voice ports. In the following example, the specified Cisco MC3810 multiservice concentrator FXS port is configured with a Bellcore/Telcordia standard (**cptone** value is **northamerica**), a station name, and a station number. The **caller-id alerting ring** setting is 1.

```
Router> show voice port 1/1
 FXS 1/1 Slot is 1, Port is 1
 Type of VoicePort is FXS
 Operation State is UP
 Administrative State is UP
 No Interface Down Failure
 Description is not set
 Noise Regeneration is enabled
 Non Linear Processing is enabled
 Music On Hold Threshold is Set to -38 dBm
 In Gain is Set to 0 dB
 Out Attenuation is Set to 0 dB
 Echo Cancellation is enabled
 Echo Cancel Coverage is set to 8 ms
 Connection Mode is normal
 Connection Number is not set
 Initial Time Out is set to 10 s
 Interdigit Time Out is set to 10 s
 Ringing Time Out is set to 180 s
 Companding Type is u-law
 Coder Type is g729ar8
 Voice Activity Detection is disabled
 Nominal Playout Delay is 80 milliseconds
 Maximum Playout Delay is 160 milliseconds
 Region Tone is set for US

 Analog Info Follows:
 Currently processing Voice
 Maintenance Mode Set to None (not in mtc mode)
 Number of signaling protocol errors are 0
 Impedance is set to 600r Ohm
 Wait Release Time Out is 30 s
 Analog interface A-D gain offset = -3.0 dB
 Analog interface D-A gain offset = -3.0 dB
 FXS idle voltage set to low

 Caller ID Info Follows:
 Standard BELLCORE
 Station name A. Person, Station number 4085551111
 Caller ID presentation unblocked
 Output attenuation is set to 14 dB
 Caller ID is transmitted after 1 rings

 Voice card specific Info Follows:
 Signal Type is loopStart
 Ring Frequency is 20 Hz
 Hook Status is Off Hook
 Ring Active Status is inactive
 Ring Ground Status is inactive
 Tip Ground Status is active
 Digit Duration Timing is set to 100 ms
 InterDigit Duration Timing is set to 100 ms
 Ring Cadence is defined by CPTone Selection
 Ring Cadence are [20 40] * 100 msec
 InterDigit Pulse Duration Timing is set to 500 ms
```

Troubleshooting Tips

If you have caller-ID problems on telephones connected to FXS ports, the following tips may be helpful:

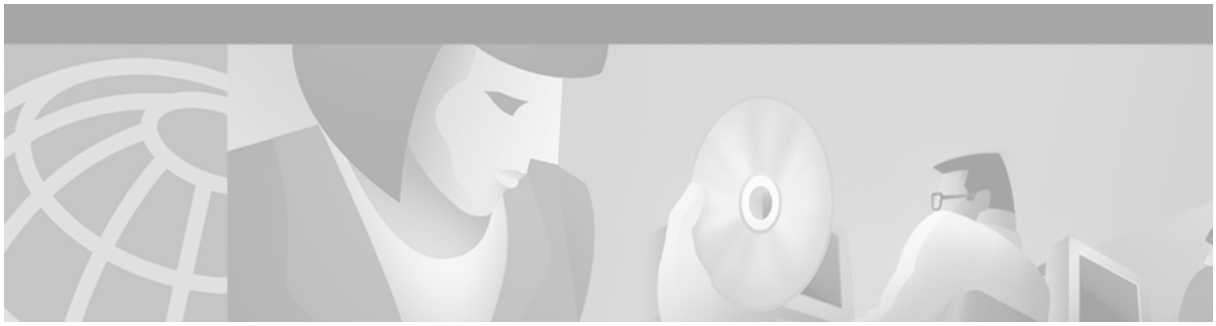
- Try a different brand of phone to confirm that the problem is not caused by a malfunctioning or incompatible caller ID telephone.
- Ensure that the **cptone** command is set correctly to reflect your locale.
- If the call time display is incorrect, check the router clock setting. An NTP network time server is recommended for accurate display of the local time.
- If expected information is not displayed, use the **show call history** command to make sure that the information that the router received during the call setup is complete.
- The line voltage available on FXS voice ports of the Cisco MC3810 multiservice concentrator and Cisco 2600 and 3600 series routers is –24V. Some phones, particularly those manufactured by Bell South, do not recognize –24V caller-ID signaling. On a Cisco MC3810 multiservice concentrator, use the **idle-voltage high** voice-port configuration command to boost the voltage on an FXS port.

If you have caller-ID display problems on FXO ports, the following tips may be helpful:

- Disconnect the router from the phone line and attach a caller-ID equipped telephone to verify that the CO is sending caller-ID information:
  - Listen and watch to see when the caller-ID information is displayed: before the first ring, after the first ring, or after the second ring?
  - Make sure that the router configuration matches the timing of the display. If the phone is answered during the first ring, does this cause the phone not to display the caller-ID information? If so, the CO may be sending the caller-ID information after the first ring, requiring a change to a caller-ID alerting setting. Make sure the router is not configured to answer the call on the FXO before the Caller ID-information is received. If needed, increase the number of rings required before answering.
- Use the **show call history** command to check the information received by the caller ID receiver.

The following **debug** commands may be useful for analyzing problems:

- **debug vpm signal**
- **debug vtsp dsp**
- **debug vtsp session**



Cisco Hoot and Holler over IP

The voice multicasting feature on Cisco 2600 and 3600 series routers uses Cisco Voice over IP (VoIP) technology to create a permanently connected point-to-multipoint hoot and holler network over an IP connection.

This appendix describes the Cisco hoot and holler over IP feature and contains the following sections:

To identify the hardware platform or software image information associated with a feature in this appendix, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the softsware release notes for a specific release. For more information, see the “Identifrying Supported Platforms” section in the “Using Cisco IOS Software” chapter.

Hoot and Holler over IP Overview

Four-wire ear and mouth (E&M), E1/T1, Foreign Exchange Office (FXO), and Foreign Exchange Station (FXS) configurations provide continuous VoIP connections across a packet network using the connection-trunk mechanism. By using the inherent point-to-multipoint connectivity of IP multicast (IPmc), the routers can take several inbound voice streams from the traditional hoot devices and forward the packetized voice over the IP network to all parties within a defined hoot and holler group.

Hoot and holler networks provide “always on” multiuser conferences without requiring that users dial into a conference. These networks came into being more than 40 years ago when local concentrations of small specialized businesses with common, time-critical informational interests began to install their own phone wires, speakers (called “squawk boxes”), and microphones between their businesses to ask each other about parts that customers needed. These networks functioned as crude, do-it-yourself, business-to-business intercom systems.

Hoot and holler broadcast audio network systems have since evolved into the specialized leased-line networks used by financial and brokerage firms to trade stocks and currency futures and the accompanying time-critical information such as market updates and morning reports.

Users of various forms of hoot and holler networks now include brokerages, news agencies, publishers, weather bureaus, transportation providers, power plant operators, manufacturers, collectibles dealers, talent agencies, and nationwide salvage yard organizations.

Hoot and holler is used in these various industries as a way to provide a one-to-many or many-to-many conferencing service for voice communications. In the past, hoot and holler was deployed using point-to-point telephone company circuits and a hoot and holler bridging and mixing functionality that was provided either by the customer or as a service of the Public Switched Telephone Network (PSTN) carrier.

A common use of hoot and holler is a broadcast audio network that is used throughout the brokerage industry to communicate morning reports as well as to advise the trading community within a brokerage firm on market movements, trade executions, and so on. All users can talk simultaneously with each other, if desired.

But more commonly, a broker in a field office will "shout" an order to the trading floor. The shout ensures that the trading floor can hear the order and a floor trader can confirm the transaction. A typical brokerage firm has several of these networks for equity, retail, and bonds with network size and degree of interactivity varying depending on the application.

Within the financial community there are two general uses for hoot and holler networks:

- Market updates—Market update (morning report) hoot networks tend to be active for an hour in the morning and inactive for the rest of the day.
- Trading—Trading hoot networks tend to be more widely used throughout the trading day.

Both of these applications can reap significant advantages by running over an IP network because any idle bandwidth can be reclaimed by data applications.

Today most hoot and holler customers pay for separate leased-line charges from a common carrier to transport their hoot and holler to remote branch offices. This recurring charge is usually significant—some larger firms spend more than $2 million to $3 million per year just to distribute hoot and holler feeds.

Cisco's hoot and holler over IP feature:

- Eliminates yearly reoccurring switched-circuit telephone company charges (toll-bypass)
- Eliminates the need for leased lines and the accompanying charges
- Reduces the need for hoot and holler bridges
- Improves hoot and holler network manageability
- Reduces the time to troubleshoot a problem from hours to minutes
- Reduces the time to provision bandwidth from days to a few hours
- Increases productivity through future applications (such as IP/TV and turret support)
- Provides the ability to integrate voice, video, and data signaling capabilities

Cisco hoot and holler over IP is supported on Cisco 2600 and 3600 series routers and on NM-HDV, NMZV, and NM-2V network modules

For information about installing voice network modules and voice interface cards in Cisco 2600 and Cisco 3600 series routers, refer to the *Cisco Network Module Hardware Installation Guide* and the *WAN Interface Card Hardware Installation Guide.*

For information about configuring Voice over IP features, refer to the *Software Configuration Guide for Cisco 3600 Series and Cisco 2600 Series Routers,* to the *Voice over IP Quick Start Guide ,* and to the "Voice over IP Overview" chapter in this configuration guide.

For further information about IP multicasting, refer to the *IP Multicast Site* at http://www.cisco.com/ipmulticast.

For further information about IP/TV, refer to the *IP/TV Content Manager User Guide.*

For further information about interactive voice response (IVR), refer to *Configuring Interactive Voice Response for Cisco Access Platforms.*

Current Hoot and Holler Implementations

Traditional hoot and holler networks (see Figure 148) are analog, multipoint, four-wire, audio-conference networks that are always up. When a user wants to communicate, the user pushes a button and speaks either through a microphone, a hoot phone, a turret, or a squawk box.

Figure 148 Traditional Hoot and Holler Network

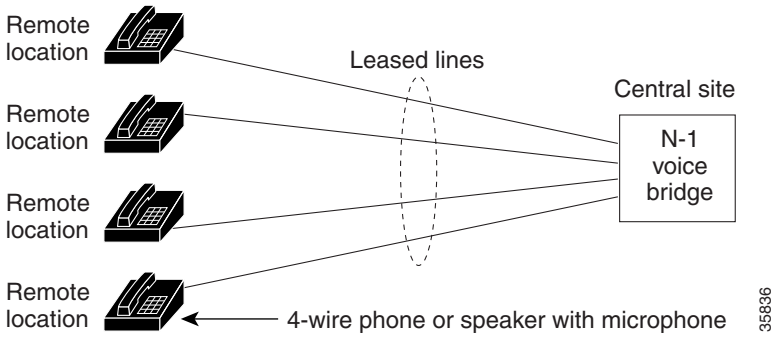


Figure 148 illustrates a traditional hoot and holler network. Each remote location is connected to a central bridge using leased lines. Four-wire connections and N-1 bridges are used to avoid echo problems.

Hoot and holler networks are typically spread over four to eight sites although financial retail networks may have hundreds of sites interconnected. Within a site, bridging (mixing voice signals) is done locally with a standard analog or digital bridge that may be part of a trading turret system. Between sites, there are two prevalent methods for providing transport:

- Point-to-point leased lines with customer-provided audio bridging at a central site
- Carrier-provided audio bridging

When customers provide their own bridging services with point-to-point leased lines, branch offices in a metropolitan area commonly have 25 to 50 lines or more.

The second method, carrier-provided audio bridging, is prevalent within the United States but rare for overseas transport. In this scenario, the audio bridges are located at the carrier's central office and the four-wire lines are terminated at the client’s site on a local audio-bridge equipped with four-wire plug-ins, which then feed to local public address (PA) system speakers. Customer-provided hoot bridging services can now be replaced with a Cisco hoot and holler over IP solution.

Cisco Hoot and Holler over IP Overview

Cisco's VoIP technology, which was initially focused on traditional PBX toll-bypass applications, can be used to combine hoot and holler networks with data networks. While some customers may have done some level of hoot and data integration in the late 1980s with time-division multiplexing (TDM), this form of integration does not allow for the dynamic sharing of bandwidth that is characteristic of VoIP.

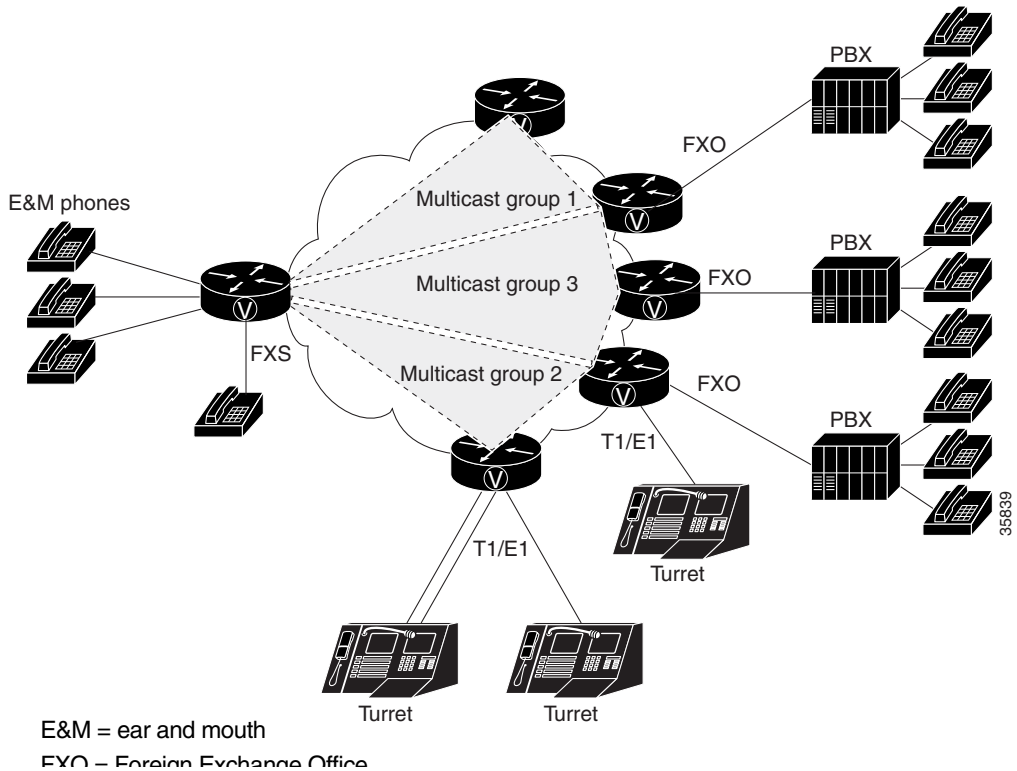
This dynamic sharing of bandwidth is even more compelling with hoot and holler than with a toll-bypass application because some hoot circuits may be active for an hour or two for morning reports but dead for the rest of the day—the idle bandwidth can be used by the data applications during these long periods of inactivity.

Beginning with Cisco IOS Release 12.1(2)XH, Cisco hoot and holler over IP can be implemented using Cisco's VoIP technology. This solution leverages Cisco's IOS expertise in VoIP, quality of service (QoS), and IP multicasting (IPmc) and is initially available on Cisco 2600 and 3600 series multiservice routers.

Figure 149 shows a diagram of the Cisco hoot and holler over IP solution connecting legacy hoot equipment over an IP network.

Note The "V" on the Cisco router icons signifies that some of the hoot and holler bridging function is being done by the router's digital signal processors (DSPs).

Figure 149 Hoot and Holler over IP Using Cisco 2600 and Cisco 3600 Series Routers



Four-wire E&M, E1/T1, FXO, and FXS configurations provide continuous VoIP connections across a packet network. By using the inherent point-to-multipoint characteristic of IPmc, the routers can take several inbound voice streams from the traditional hoot devices and forward the packetized voice over the IP network to all parties within a defined hoot and holler group.

Voice Multicasting

The voice multicasting feature on Cisco 2600 and Cisco 3600 series routers uses Cisco VoIP technology to create a point-to-multipoint hoot and holler network over an IP connection.

You can connect voice multicasting telephones to routers in the following ways:

- Connect a four-wire E&M telephone, which has no dial and is always off-hook, directly to an E&M voice interface card that is installed in a voice network module. Configure the E&M interface for four-wire trunk operation. For information about configuring E&M interfaces, see the chapter "Configuring Voice Ports" in this configuration guide.
- Connect a conventional telephone to a PBX that is connected to an E&M voice interface card.
- Connect a conventional telephone to an FXS voice interface card that is installed in a voice network module.
- Connect a conventional telephone to a PBX that is connected through a E1/T1 line to a multiflex trunk interface card that is installed in a high-density voice network module.

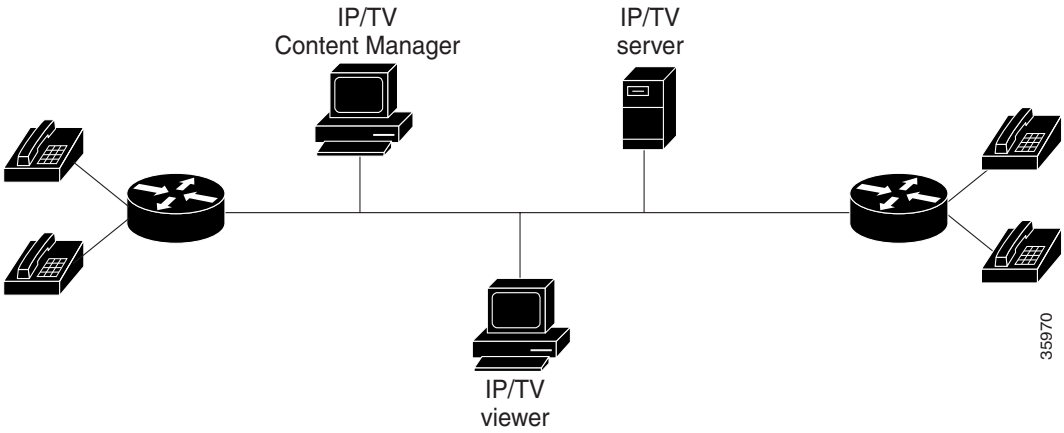
Note The voice multicasting feature supports only one E1/T1 line per high-density voice network module.

IP/TV Access

The Cisco hoot and holler over IP feature enables you to access ongoing IP/TV multicasts for listening to voice content of the IP/TV session. For complete information on IP/TV, see the *IP/TV Content Manager Installation and User Guide*.

The following figure illustrates Cisco hoot and holler being used to access IP/TV multicasts:

Figure 150 Cisco Hoot and Holler over IP Access to IP/TV Multicast



For the Cisco hoot and holler over IP and IP/TX interaction to work correctly, do the following:

- Ensure that you have properly connected and configured your network for VoIP. Enable the Cisco hoot and holler over IP feature using the **session protocol multicast** command.
- Ensure that the server configured with the IP/TV Content Manager is in the same Ethernet network as the Cisco hoot and holler over IP functionality.
- Ensure that the Cisco hoot and holler over IP multicast details are registered with the IP/TV Content Manager.

Note IP/TV support for Cisco hoot and holler over IP uses only G.711 u-law (mu-law) encoding.

IP/TV supports one audio stream for Cisco hoot and holler over IP.

IP/TV does not support arbitration and mixing.

Content Manager

On the configuration screen (Administration Tool>Scheduled Programs>New Program>Configuration), provide the following details:

- Multicast address
- RTP port—defined by the dial peer in the router
- IP/TV server—IP address or name
- From the Settings>Content Manager option, do the following:
  - Click Add New.
  - Enter the IP/TV server name.
  - Enter the port number. It must be 80 because it is HTTP.
  - Click OK and exit.

Note In Content Manager, be sure to specify the multicast IP address and RTP port for the Cisco hoot and holler over IP session.

Interactive Voice Response

The Cisco hoot and holler over IP feature can support interactive voice response (IVR) as a means of authentication, authorization, and accounting (AAA) control. See the "Configuring TCL IVR Applications" chapter in this configuration guide or refer to the *Cisco IOS Voice, Video, and Fax Command Reference* for more information.

Migration Strategy

To aid troubleshooting and allow for regionalized hoot and holler conferences, most hoot and holler networks today are structured by interconnecting multiple regional hoot networks with a centralized bridge. The regional hoot networks are built using either carrier-based multidrop circuits or point-to-point circuits bridged by the customer. All of these circuits are connected through patch panels that allow for these regional bridges to be connected for a larger corporate-wide conference call. This is typically done for the "morning call" that is broadcast to all locations, advising of market movements, recommendations, and commentary. Later in the day, the patch panel may be reconfigured to allow for local or regional conference bridges. This allows for multiple conference calls for various purposes, without provisioning multiple circuits. By segmenting the network into regions, troubleshooting is also easier because any audio disturbance, feedback, or level problems can be isolated to a smaller subset of remote offices for more specific troubleshooting.

The highly segmented nature of existing hoot and holler networks can be leveraged in the migration from legacy hoot technology to Cisco hoot and holler over IP. A small segment of the hoot network can be converted to Cisco hoot and holler over IP while preserving the operational procedures at the main office.

Note that the migration to Cisco hoot and holler over IP does not require replacing end-user equipment or central bridging equipment. The main impetus for this first phase of migration is to eliminate the recurring expense of carrier multidrop circuits or dedicated leased lines. Migration success is maximized by minimizing changes to the end user while realizing an attractive payback period on the capital costs.

As the entire hoot network converges with the data network, additional functionality can be introduced. Since the hoot and holler connections are now carried in standard multicast RTP packets, hoot channels can now be received by a soft client such as IP/TV, which can receive an IP multicast RTP stream. An alternate migration strategy is to use Cisco hoot and holler over IP technology initially as a backup for the existing hoot circuits within a region with a phased plan of cutting over to Cisco hoot and holler over IP as the primary transport while keeping the existing circuits as a backup for a predefined burn-in period.

Technical Details of the Cisco Hoot and Holler over IP Solution

This section describes how Cisco hoot and holler over IP works from a technical perspective. It covers design considerations in terms of IOS configurations and DSP mixing functionality. It also covers bandwidth planning and QoS, with the following assumptions:

- That you have some level of Cisco IOS experience.
- That you have some experience configuring QoS features using Cisco IOS software. For assistance, refer to the *Cisco IOS IP and IP Routing Configuration Guid*e at the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/index.htm
- That you have some experience configuring VoIP using Cisco IOS software. For assistance, refer to the *Cisco IOS Voice over IP Overview* at the following URL:
- http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fvvfax_c/vvfvoip.htm
- That you have some experience configuring IP multicasting using Cisco IOS software. For assistance, refer to *Cisco IOS Configuring IP Multicast Routing* at the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcpt3/index.htm
- That you have a working IP network, with IP multicasting configured using the Cisco 2600 and Cisco 3600 series routers. For assistance, refer to the following documents at the Cisco Connection Online (CCO) Web site:
  - *Cisco IOS Configuration Guides and Command References* (http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/index.htm)
  - *Cisco 2600 Series Routers* (http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis2600/index.htm)
  - *Cisco 3600 Series Routers* (http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/index.htm)
- That you are familiar with Cisco IP/TV. For assistance, refer to *Cisco IOS Software Configuration* at the following URL:

  http://www.cisco.com/univercd/cc/td/doc/product/software/iptv30/
- That you understand basic hoot and holler concepts and equipment.

IP Multicast and DSP Arbitration and Mixing

When deploying Cisco hoot and holler over IP, first consider how the voice streams are going to be mixed and how they will be distributed to other locations. This is done using a combination of two technologies:

- IPmc
- DSP arbitration and mixing

Since hoot and holler is generally used to allow many people to simultaneously talk and listen to other people within a hoot group, by definition it requires that the same speech be delivered to multiple parties at the same time. In an IP network, this functionality uses IPmc. IPmc allows a source to send a single packet into the IP network and have it duplicated and sent to many listeners by the other routers within the network. This technique is beneficial in that it does not require the source to know how many listeners there are, and the source does not have the additional processing burden of having to send a copy of each packet to all listeners. IPmc also allows listeners to dynamically join IPmc groups, which eliminates the administrative burden of new users needing to be added every time a new IPmc session is initiated.

The individual router/gateway can handle mixing and arbitrating the various voice streams that can be initiated or terminated on its voice ports. This functionality is handled by the onboard DSPs on each voice card (NM-1V, NM-2V, or NM-HDV). Arbitration involves identifying the various sources of the voice stream, and mixing involves taking some of those voice streams and combining them into a single-sourced voice stream. Cisco hoot and holler over IP can handle many inbound voice streams, *but it only arbitrates and mixes three streams to be heard within the hoot group*. This value works fine in most applications because, with more than three streams, two things happen in normal conversation:

- People are not able to distinguish the content of more than three voice streams.
- People normally stop speaking if they hear others talking ahead of them.

Note The mixing functionality does not do a summation of the voice streams.

The DSPs in the Cisco hoot and holler over IP feature can mix up to three streams. The mixing of the three streams is important to network administrators in considering how much bandwidth they should plan for in their Cisco hoot and holler over IP networks. This is especially crucial when planning for WAN bandwidth, which is often much more expensive and much less available than LAN bandwidth.

The advantage to this functionality is that a network administrator never has to be concerned about provisioning voice bandwidth for more than three times each call's bandwidth for each WAN site, which helps to simplify overall network planning.

Bandwidth Planning

Four main factors must be considered with regard to bandwidth planning for Cisco hoot and holler over IP:

- Codecs used for VoIP (G.711, G.726, G.729, and G.729a are currently supported).
- Bandwidth management techniques.
- The number of voice streams to be mixed.
- The amount of guaranteed bandwidth available on the IP network. This includes both LAN and WAN bandwidth and should take into consideration other factors, such as Frame Relay committed information rate.

Codecs

By default, Cisco IOS sends all VoIP traffic (media, using RTP) at a rate of 50 packets per second. The packets include not only the voice sample, but also an IP, User Datagram Protocol (UDP), and RTP header. The IP/UDP/RTP header adds an additional 40 bytes to each packet. The amount of bandwidth each VoIP call consumes depends on the codec selected. The resulting bandwidths can be as follows:

- G.729 or G.729a = 3,000 bytes * 8 bits = 24 Kb/call
- G.726 = 6,000 bytes * 8 bits = 48 Kb/call
- G.711 = 10,000 bytes * 8 bits = 80 Kb/call

In addition to these calculations, network administrators must consider the Layer 2 headers (Frame Relay, PPP, Ethernet, and so on) and add the appropriate number of bytes to each packet.

In Table 61, the assumption is that the payload size (in bytes) is 20-millisecond samples per packet with 50 packets per second.

The value of *n* is equal to the number of voice streams in a session.

The uncompressed bandwidth includes IP/UDP/RTP headers (40 bytes) in the bandwidth calculation. Compressed RTP (cRTP) reduces the IP/UDP/RTP headers to between 2 to 4 bytes per packet. The calculation of compressed bandwidth below uses 4 bytes for a compressed IP/UDP/RTP header per packet.

Maximum RTP Control Protocol (RTCP) bandwidth is 5 percent of the total RTP traffic in a hoot and holler session. Since the Cisco hoot and holler over IP application supports mixing of a maximum of three voice streams, the RTCP bandwidth is limited to 5 percent of three-voice-stream traffic.

In addition to the above, Layer 2 headers (Frame Relay, Point-to-Point Protocol, Ethernet, and so on) should be considered and added to the bandwidth calculation.

Table 61 Bandwidth Consideration Table

Codec	Payload Size (byte)	Bandwidth/ Voice Stream (Kbps)		RTCP Bandwidth per Cisco Hoot and Holler over IP Session (Kbps)	Example—One Voice Stream in a Session (Bandwidth in Kbps)	
		Uncompressed	Compressed		=(1)*n+(3)	=(2)*n+(3)
g.729	20	24	9.6	3.6	27.6	13.2
g.726	80	48	33.6	7.2	55.2	40.8
g.711	160	80	65.6	12.0	92.0	77.6

cRTP, Variable-Payload Sizes and VAD

Some network administrators may consider this amount of bandwidth per call unacceptable or outside the limits of the bandwidth they can provide, especially in the WAN. There are several options that network administrators have for modifying the bandwidth consumed per call:

- RTP header compression (cRTP)
- Adjustable byte size of the voice payload
- Voice activity detection (VAD)

IP/UDP/RTP headers add an additional 40 bytes to each packet, but each packet header is basically unchanged throughout the call. Compressed RTP can be enabled for the VoIP calls, which reduces the IP/UDP/RTP headers from 2 bytes to 4 bytes per packet.

More information on cRTP may be found in the "Quality of Service Overview" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide* at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/qcfintro.htm

In addition to reducing the IP/UDP/RTP headers per packet, the network administrator also has the option of controlling how much voice payload is included in each packet. This is done using the **bytes** keyword and argument in a VoIP dial peer. The following example shows a dial-peer configuration:

```
dial-peer voice 1 voip
 destination-pattern 4085551234
 codec g729r8 bytes 40
 session protocol multicast
 session target ipv4:239.10.108.252:20102
```

As the number of bytes per packet is modified, so too is the number of packets per second that are sent.

VAD enables the DSPs to dynamically sense when there are pauses in a conversation. When these pauses occur, no VoIP packets are sent into the network. This significantly reduces the amount of bandwidth used per VoIP call, sometimes as much as 40 percent to 50 percent. When voice is present, VoIP packets are again sent. When using Cisco hoot and holler over IP, VAD must be enabled to reduce the amount of processing of idle packets by the DSPs. In basic VoIP, VAD can be enabled or disabled, but since the DSPs also have to do arbitration and mixing, VAD must be disabled to reduce the DSPs' processing load. In addition to enabling VAD (which is only by default), network administrators should modify the VAD parameters that sense background noise so that idle noise does not consume bandwidth.

This can be configured as in the following E&M port example:

```
 voice class permanent 1
 signal timing oos timeout disabled
 signal keepalive 65535
!
voice-port 1/0/0
 voice-class permanent 1
 connection trunk 111
 music-threshold -30
 operation 4-wire
```

The configuration above is used for a voice port that is in send/receive mode, and only noise above -30Db is considered voice.

Virtual Interface

In all Cisco hoot and holler over IP implementations, the routers are configured with an "interface vif1." This is a virtual interface that is similar to a loopback interface—a logical IP interface that is always up when the router is active. In addition, it must be configured so the Cisco hoot and holler over IP packets that are locally mixed on the DSPs can be fast-switched along with the other data packets. This interface must reside on its own unique subnet, and that subnet should be included in the routing protocol updates (Routing Information Protocol [RIP], Open Shortest Path First [OSPF], and so on).

Connection Trunk

Cisco hoot and holler over IP provides an "always-on" communications bridge—end users do not need to dial any phone numbers to reach the other members of a hoot group. To simulate this functionality, Cisco IOS provides a feature called "connection trunk." Connection trunk provides a permanent voice call, without requiring any input from the end user, because all the digits are internally dialed by the router/gateway.

With traditional VoIP usages of connection trunk, the call is mapped to a remote router/gateway, and all the H.323 signaling is handled dynamically when the trunk is established. With hoot and holler over IP, the connection trunk is associated with the IP address of the IP multicast group that maps to the hoot group.

In addition, all negotiation of UDP ports for the audio stream is manually configured. The following example shows an E&M voice port connection trunk set up for Cisco hoot and holler over IP:

```
voice-port 1/0/0
 connection trunk 111
 music-threshold -30
 operation 4-wire
!
dial-peer voice 1 voip
 destination-pattern 111
 voice-class permanent 1
 session protocol multicast
 session target ipv4:237.111.0.0:22222
 ip precedence 5
```

In this example, the digits in the **connection trunk** *111* string match the destination pattern of the VoIP dial peer. Also, the session protocol is set to multicast and the session target is pointing to the IPmc group number, with the UDP port (22222) predefined.

Cisco Hoot and Holler over IP Restrictions

The restrictions for using Cisco hoot and holler over IP are as follows:

- Cisco hoot and holler over IP supports the mixing of only three voice streams.
- IP/TV does not support the mixing of audio streams.
- IP/TV supports only G.711 u-law (mu-law).
- Voice Interface Card Basic Rate Interface (VIC-BRI) is not supported.

Configuration Tasks

To configure Cisco hoot and holler over IP, perform the tasks in the following sections:

- Configuring Multicast Routing, page 876 (Required)
- Configuring the Virtual Interface, page 876 (Required)
- Configuring VoIP Dial Peers, page 877 (Required)
- Configuring E&M Voice Ports, page 879 (Required, if used)
- Configuring for Receive Only Mode, page 881 (Optional)
- Configuring Relevant Interface (Serial/Ethernet), page 882 (Required)
- Configuring Voice Ports in High-Density Voice Network Modules, page 882 (Required, if using T1/E1)

Configuring Multicast Routing

To enable multicast routing, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# ip multicast-routing`	Enables multicast routing.

Configuring the Virtual Interface

To configure the virtual interface for multicast fast switching, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *type number* [*name-tag*]	Defines a virtual interface for multicast fast switching. Routers joining the same session must have their virtual interfaces on different subnets. Otherwise, packets are not switched to the IP network.
Step 2	`Router(config-if)# ip address` ip-*address mask* [**secondary**]	Assigns the IP address and subnet mask for the virtual interface.
Step 3	`Router(config-if)# ip pim` {**dense-mode** | **sparse mode** | **sparse-dense-mode**}	Specifies Protocol Independent Multicast (PIM). Whatever mode you choose should match all the interfaces in all the routers of your network.

Configuring VoIP Dial Peers

To configure the VoIP dial peers on the router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *tag* `voip`	Assigns a variable number (*tag*) to the VoIP dial peer and enters dial-peer configuration mode.
Step 2	`Router(config-dial-peer)# destination-pattern [+]`*string*`[T]`	Specifies the E.164 address associated with this dial peer. The destination pattern for the VoIP dial peer must match the value of the *multicast-session-number* string for the corresponding voice port. The keywords and arguments are as follows: • **+**—(Optional) Specifies a character indicating an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810. • *string*—Indicates a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – The asterisk (*) and pound sign (#)—Indicate the keys that appear on standard touch-tone dial pads. On the Cisco 3600 series only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,)—Inserts a pause between digits. – Period (.)—Matches any entered digit (this character is used as a wildcard). On the Cisco 3600 series, the period cannot be used as a leading character in a string (for example, .650). – Percent sign (%)—Indicates that the previous digit/pattern occurred zero or multiple times, similar to the wild card usage in the regular expression.

Command	Purpose
	– Plus sign (+)—Matches a sequence of one or more matches of the character/pattern. **Note** The plus sign used as part of the digit string is different from the plus sign that can be used in front of the digit string to indicate that the string is an E.164 standard number. – Circumflex (^)—Indicates a match to the beginning of the string. – Dollar sign ($)—Matches the null string at the end of the input string. – Backslash symbol (\)—Is followed by a single character matching that character or used with a single character having no other significance (matching that character). – Question mark (?)— Indicates that the previous digit occurred zero or one time. – Brackets ([])—Indicates a range of digits. A range is a sequence of characters enclosed in the brackets, and only numeric characters from "0" to "9" are allowed in the range. This is similar to a regular expression rule. – Parentheses ( )—Indicates a pattern and is the same as the regular expression rule—for example, 408(555). Parentheses are used in conjunction with symbols ?, %, or +. For more information on applying wildcard symbols to destination patterns and the dial strings that result, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide. • **T**—(Optional) Control character indicating that the **destination-pattern** value is a variable length dial string.
Step 3 `Router(config-dial-peer)# session protocol multicast`	This step is mandatory for voice multicasting and is the command introduced specifically for the Cisco hoot and holler over IP application.

	Command	Purpose
Step 4	`Router(config-dial-peer)# session target`	Assigns the session target for voice multicasting dial peers. This is a multicast address in the range of from 224.0.1.0 to 239.255.255.255 and must be the same for all ports in a session. The audio RTP port is an even number in the range of from 16384 to 32767 and must also be the same for all ports in a session. An odd-numbered port (UDP port number + 1) is used for the RTCP traffic for that session.
Step 5	`Router(config-dial-peer)# ip precedence number`	(Optional) Specifies the IP precedence.
Step 6	`Router(config-dial-peer)# codec`	Configures the codec. You must configure the same codec on all dial peers in a session. When the default codec **g729r8** is used, it does not appear in the configuration when the **show running-config** command is used.

Configuring E&M Voice Ports

To configure E&M voice ports, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class permanent tag`	Defines voice class for transmit-receive mode.
Step 2	`Router(config-class)# signal timing oos timeout seconds disabled`	Disables signaling loss detection. (Use the **disabled** keyword in hoot and holler applications. The *seconds* argument is not used in these applications.)
Step 3	`Router(config-class)# signal keepalive number`	Specifies the keepalive signaling packet interval.
Step 4	`Router(config-class)# exit`	Exits voice-class configuration mode.
Step 5	`Router(config)# voice-port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Enters voice-port configuration mode and selects the voice port to configure.
Step 6	`Router(config-voice-port)# voice-class permanent tag`	Assigns a previously configured voice class for a Cisco trunk or FRF.11 trunk to a voice port (for the port that is allowed to speak).
Step 7	`Router(config-voice-port)# connection {plar | tie-line | plar-opx} digits | {trunk digits [answer-mode]}`	Ties the voice port to a multicast-session number. Use the **trunk** keyword to specify a connection that emulates a permanent trunk connection to a PBX. The *digits* argument specifies the destination telephone number. Valid entries are any series of digits that specify the E.164 telephone number.
Step 8	`Router(config-voice-port)# music-threshold number`	(Optional) Sets the music threshold to make voice-activated dialing (VAD) less sensitive. The *number* argument is the on-hold music threshold in decibels (dB). Valid entries are any integer from -70 to -30.

	Command	Purpose
Step 9	`Router(config-voice-port)# operation {2-wire | 4-wire}`	Specifies the cabling scheme for E&M ports. The **2-wire** keyword is the default. (Choose **4-wire** operation for the hoot and holler application.)
Step 10	`Router(config-voice-port)# type {1 | 2 | 3 | 5}`	Selects the appropriate E&M interface type (depending on the end connection—such as PBX): • Type 1 indicates the following lead configuration (default—this is the recommended option): – E—Output, relay to ground – M—Input, referenced to ground • Type 2 indicates the following lead configuration: – E—Output, relay to SG – M—Input, referenced to ground – SB—Feed for M, connected to -48V – SG—Return for E, galvanically isolated from ground • Type 3 indicates the following lead configuration: – E—Output, relay to ground – M—Input, referenced to ground – SB—Connected to -48V – SG—Connected to ground • Type 5 indicates the following lead configuration: – E—Output, relay to ground – M—Input, referenced to -48V
Step 11	`Router(config-voice-port)# signal {wink-start | immediate | delay-dial}`	Configures the signaling type for E&M voice ports. The default is **wink-start**. Select **immediate** for the Cisco hoot and holler over IP application. In the immediate-start protocol, the originating side does not wait for a wink before sending addressing information. After receiving addressing digits, the terminating side then goes off-hook for the duration of the call. The originating endpoint maintains off-hook for the duration of the call.
Step 12	`Router(config-voice-port)# voice-port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Selects another voice port.
Step 13	`Router(config-voice-port)# voice-class permanent tag`	Uses the voice class *tag* for the receive-only port.

	Command	Purpose
Step 14	`Router(config-voice-port)# connection {plar | tie-line | plar-opx} digits | {trunk digits [answer-mode]}`	Ties the voice port to the same multicast-session number as in Step 12. (Use the **trunk** keyword and the *digits* argument for the hoot and holler application.)
Step 15	`Router(config-voice-port)# music-threshold number`	(Optional) Sets the music threshold to make VAD less sensitive.
Step 16	`Router(config-voice-port)# operation {2-wire | 4-wire}`	Specifies the calling scheme for E&M ports. The **2-wire** keyword is the default. (Specify **4-wire** operation for the hoot and holler application.)

Configuring for Receive Only Mode

To configure Cisco hoot and holler over IP as receive-only mode, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config-class)# voice class permanent tag`	Enters voice-class configuration mode and defines the voice class for receive-only mode.
Step 2	`Router(config-class)# signal pattern {idle receive | idle transmit | oos receive | oos transmit} word`	Specifies the received signal pattern. Configures the ABCD bit pattern for Cisco trunks and FRF.11 trunks. (Specify the **oos receive** keywords and the *word* argument for the hoot and holler application.)
Step 3	`Router(config-class)# signal timing oos suppress-all seconds`	If the transmit out-of-service pattern (from the PBX to the network) matches for the time specified, the router stops sending packets to the network.
Step 4	`Router(config-class)# signal keepalive number`	Specifies keepalive signaling packet interval.

Configuring Relevant Interface (Serial/Ethernet)

To configure either the serial or Ethernet interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type slot/port-adapter/port [ethernet | serial]`	Configures the physical interface (serial/Ethernet) for transmitting multicast packets and enters interface configuration mode.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Assigns the IP address and subnet mask for the interface. The **secondary** keyword is optional. It specifies that the configured address is a secondary IP address. If this keyword is omitted, the configured address is the primary IP address.
Step 3	`Router(config-if)# ip pim {dense-mode | sparse mode | sparse-dense-mode}`	Specifies Protocol Independent Multicast (PIM). Whatever mode you choose should match all the interfaces in all the routers of your network.
Step 4	`Router(config-if)# no shutdown`	Enables the interface.

Configuring Voice Ports in High-Density Voice Network Modules

A multiflex trunk interface card (NM-HDV) in a high-density voice network module requires special voice-port configuration when connecting for T1/E1 operation. To configure a multiflex trunk interface card in a high-density voice network module, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# voice class permanent tag`	Defines voice class for transmit-receive mode and enters voice class configuration mode.
Step 2	`Router(config-class)# signal timing oos timeout seconds | disabled`	Disables signaling loss detection. (Use the **disabled** keyword for hoot and holler applications.)
Step 3	`Router(config-class)# signal keepalive number`	Specifies the keepalive signaling packet interval in seconds. The valid range is from 1 to 65535.
Step 4	`Router(config-class)# exit`	Exits voice class configuration mode.
Step 5	`Router(config)# voice-card slot`	Enters voice-card configuration mode and selects the card to configure. The *slot* argument is a value from 0 to 3 that identifies the physical slot in the chassis where the voice card is located.
Step 6	`Router(config-voicecard)# codec complexity {high | medium}`	Specifies call density and codec complexity based on the codec standard you are using. For hoot and holler applications, the codec complexity must be **high**. Voice multicasting does not support medium complexity, which is the default.
Step 7	`Router(config-voicecard)# exit`	Exits voice-card configuration mode.

	Command	Purpose
Step 8	`Router(config)# controller {t1 | e1} number`	Enters controller configuration mode and selects the T1 or E1 controller to configure.
Step 9	`Router(config-controller)# ds0-group ds0-group-no timeslots timeslot-list type type`	Maps a group of time slots to a DS0 group.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 11	`Router(config)# voice-port {slot-number/subunit-number/port} | {slot/port:ds0-group-no}`	Enters voice-port configuration mode and configures a DS0 group that was created in Step 9 in this configuration task table.
Step 12	`Router(config-voice-port)# connection {plar | tie-line | plar-opx} digits | {trunk digits [answer-mode]}`	Ties the connection trunk to a multicast address. This command is repeated for each DS0 group. All groups use the same multicast address if connecting to the same multicast session. (Use the **trunk** keyword and the *digit* argument for hoot and holler applications.)
Step 13	`Router(config-voice-port)# voice-class permanent tag`	Specifies the receive-only port.

Configuration Examples

This section provides a series of configuration examples to help you become familiar with voice multicasting. These examples also show how to ensure that each configuration is working properly before proceeding to the next step.

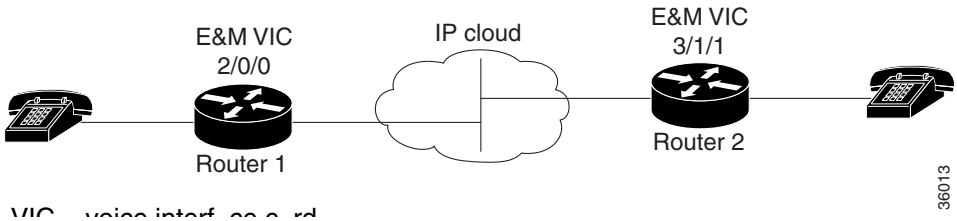
- Voice Multicasting over an Ethernet LAN, page 884
- Voice Multicasting over a WAN, page 887
- Cisco Hoot and Holler over IP with Ethernet Topology (Two Hoot Groups)
- Cisco Hoot and Holler over IP with Frame-Relay Topology (One Hoot Group)

Note In all the following configuration examples, the routers are configured with the **interface** command and the name tag **vif1**. This is a virtual interface that is similar to a loopback interface—it is a logical IP interface that is always up when the router is active. In addition, it must be configured so that the Cisco hoot and holler over IP packets that are locally mixed on the DSPs can be fast-switched along with the other data packets. This interface needs to reside on its own unique subnet, and that subnet should be included in the routing protocol updates (RIP, OSPF, and so on).

Voice Multicasting over an Ethernet LAN

Figure 151 shows the simplest configuration for voice multicasting over an Ethernet LAN. Two routers are connected to each other over the Ethernet LAN. One E&M phone is connected to each router.

Figure 151 Voice Multicasting over a LAN



In router Abbott (Figure 151), the phone is connected to voice port 2/0/0, using the *router-slot/voice-slot/VIC-port* numbering convention. This router is configured as in the following example:

```
hostname Abbott
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
 ip address 1.1.1.1 255.0.0.0
 ip pim sparse-dense-mode
!
interface Ethernet0/0
 ip address 3.3.3.1 255.0.0.0
 ip pim sparse-dense-mode
!
ip route 2.0.0.0 255.0.0.0 3.3.3.2
!
voice-port 2/0/0
 voice-class permanent 1
 connection trunk 111
 operation 4-wire
!
dial-peer voice 1 voip
 destination-pattern 111
 session protocol multicast
 session target ipv4:237.111.0.111:22222
!
```

Note The connection-trunk connection type is a point-to-point connection, similar to a tie-line on a PBX network. All voice traffic—including signaling—placed at one end is immediately transferred to the other.

Note The E&M voice port must be configured for four-wire operation.

Configuring the Second Router

In router Costello (Figure 151), the E&M phone is connected to voice port 3/1/1. Router Costello uses the same configuration as Abbott, except for the following differences:

- The virtual interface must be on a different subnet from the first router.
- The IP address in the Ethernet configuration must be different from that of the first router.
- The voice port and slot should match the router's hardware configuration.

```
hostname Costello
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
 ip address 2.2.2.2 255.0.0.0
 ip pim sparse-dense-mode
!
interface Ethernet0/0
 ip address 3.3.3.2 255.0.0.0
 ip pim sparse-dense-mode
!
ip route 1.0.0.0 255.0.0.0 3.3.3.1
!
voice-port 3/1/1
 voice-class permanent 1
 timeouts wait-release 3
 connection trunk 222
 music-threshold -30
 operation 4-wire
!
dial-peer voice 1 voip
 destination-pattern 111
 session protocol multicast
 session target ipv4:237.111.0.111:22222
!
```

Note The multicast session for this port, shown in the **session target** command, must match the multicast session configured on the first router.

The codec configured for this dial peer must match the codec for the dial peer on the first router.

Both routers must be configured to use the same connection trunk and destination pattern.

Verifying the Configuration

If you have configured your routers by following these examples, you should now be able to talk over the telephones. You can also use the **show dial-peer voice** command on each router to verify that the data you configured is correct.

To verify that an audio path has been established, use the **show call active voice** command. This command displays all active voice calls traveling through the router.

High-Density Voice Modules

A multiflex trunk interface card in a high-density voice network module requires special voice-port configuration. The card must be configured first as is shown in the following output:

```
voice-card 6
 codec complexity high
!
```

Note Codec complexity must be high. Voice multicasting does not support medium complexity, which is the default.

The following commands show how to define the T1 channel and signaling method and how to map each DS0 to voice port *slot/port:ds0-group-no*:

```
controller T1 6/0
 ds0-group 1 timeslots 1 type e&m-immediate-start
 ds0-group 2 timeslots 2 type e&m-immediate-start
 ds0-group 3 timeslots 3 type e&m-immediate-start
 …
 ds0-group 22 timeslots 22 type e&m-immediate-start
 ds0-group 23 timeslots 23 type e&m-immediate-start
```

The following commands show how to configure the voice ports on the multiflex trunk interface card:

```
!
voice-port 6/0:1
 connection trunk 111
!
voice-port 6/0:2
 connection trunk 111
!
voice-port 6/0:3
 connection trunk 111
 …
voice-port 6/0:22
 connection trunk 111
!
voice-port 6/0:23
 connection trunk 111
```

Dial-Peer Configuration

Cisco IOS software uses dial peers to tie together telephone numbers, voice ports, and other call parameters. Configuring dial peers is similar to configuring static IP routes—you are instructing the router what path to follow to route the call.

Dial peers are identified by numbers, but to avoid confusing these numbers with telephone numbers, they are usually referred to as tags. Dial peer tags are integers that can range from 1 to 2^{31} -1 (2147483647). Dial peers on the same router must have unique tags, but you can reuse the tags on other routers.

The following commands show how to configure a dial peer with tag 1 for this voice port:

```
!Configure dial peer.
!Conference 1.
!Phone number 111.
!Multicast address 237.111.0.0, udp port 22222.
dial-peer voice 1 voip
 destination-pattern 111
 session protocol multicast
```

```
 session target ipv4:237.111.0.0:22222
 ip precedence 5
 codec g711ulaw
!
```

Note The configuration for the **codec g711ulaw** in the above configuration is not necessary—the default codec of **g729r8** could be used (and it would not display for **show config**).

Tips

- The **destination-pattern** *111* for the VoIP dial peer matches the connection trunk string for the corresponding voice port.
- The **session protocol multicast** command is essential for voice multicasting.
- The session target for voice multicasting dial peers is a multicast address in the range of from 224.0.1.0 to 239.255.255.255. *This session target must be the same for all routers in a session.* The audio RTP port is an even number in the range of from 16384 to 32767 and must also be the same for all routers in a session. An odd-numbered port (UDP port number + 1) is used for the RTCP traffic for that session.
- The following codec restrictions apply:
  - You must configure the same codec on all dial peers and routers in a session.
  - Only G.711, G.726, and G.729 codecs are supported.
  - When the default codec, G.729r8, is used, it does not appear in the configuration.
- Voice activity detection is enabled by default. Cisco recommends that this setting should not be changed.

Ethernet Configuration

Configure the router's Ethernet interface as follows:

```
!Configure physical interface for transmitting multicast packets.
!
interface ethernet 0/0
 ip address 1.5.13.13 255.255.255.0
 ip pim sparse-dense-mode!
```

Voice Multicasting over a WAN

The configuration for voice multicasting sessions over IP on Frame Relay is the same as for the Ethernet LAN in the previous example. Configure the WAN interface on each router with the **ip address** and **ip pim** commands and the **sparse-dense-mode** keywords as shown in the section "Voice Multicasting over an Ethernet LAN."

Quality of Service

Voice traffic is much more sensitive to timing variations than data traffic. For good voice performance, configure your data network so that voice packets are not lost or delayed. The following example shows one way to improve QoS for voice multicasting over a Frame Relay connection:

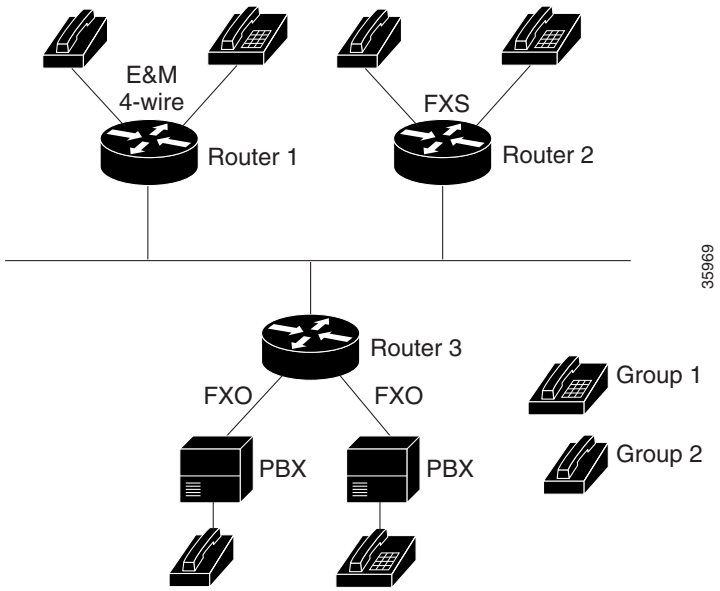
```
!Configure physical interface for transmitting multicast packets.
!Listen to packets of Session Announcement Protocol (SAP).
!This example uses a subinterface
!
interface serial0/0
 encapsulation frame-relay
 frame-relay traffic-shaping
 no frame-relay broadcast-queue
!
interface serial0/0.1 point-to-point
 ip address 5.5.5.5 255.255.255.0
 ip pim sparse-dense-mode
 frame-relay class hootie
 frame-relay interface-dlci 100
 frame-relay ip rtp header-compression
!
!Frame relay class commands.
!
map-class frame-relay hootie
 frame-relay cir 64000
 frame-relay bc 2000
 frame-relay mincir 64000
 no frame-relay adaptive-shaping
 frame-relay fair-queue
 frame-relay fragment 80
 frame-relay ip rtp priority 16384 16383 64
```

Note In the **frame-relay ip rtp priority** command, the first number is the audio port. The second number is the number of consecutive audio ports to which the IP RTP priority queuing applies. The third number is the bandwidth, which should equal the bandwidth needed for each call multiplied by the number of calls.

Cisco Hoot and Holler over IP with Ethernet Topology (Two Hoot Groups)

The following figure illustrates Cisco hoot and holler over IP being used with an Ethernet topology:

Figure 152 Cisco Hoot and Holler over IP with Ethernet Topology



In this configuration, two hoot and holler groups are set up by defining two multicast groups (237.16.8.11 and 237.16.26.12) and mapping the **connection** command and **trunk** keyword (specifying *11)* and **connection** command and **trunk** keyword (specifying *12)* from the voice ports to the VoIP dial peers associated with each group. Each router is connected to a dedicated switch port, and IP precedence is set to 5 for all Cisco hoot and holler over IP packets.

Router-1 (E&M Four-Wire Ports)

The following output shows that Router 1 has been configured for E&M four-wire ports:

```
hostname Router-1
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
ip address 1.1.1.1 255.255.255.0
ip pim sparse-dense-mode
!
interface Ethernet0/0
ip address 1.5.13.1 255.255.255.0
ip pim sparse-dense-mode
!
router rip
network 1.1.1.0
network 1.5.13.0
!
```

```
voice-port 1/0/0
voice-class permanent 1
connection trunk 111
music-threshold -30
operation 4-wire
!
voice-port 1/0/1
voice-class permanent 1
connection trunk 112
music-threshold -30
operation 4-wire
!
dial-peer voice 111 voip
destination-pattern 111
session protocol multicast
session target ipv4:237.111.0.111:22222
ip precedence 5
!
dial-peer voice 112 voip
destination-pattern 112
session protocol multicast
session target ipv4:239.194.0.10:22224
ip precedence 5
!
```

Router-2 (FXS Ports)

The following output shows that Router 2 has been configured for FSX ports:

```
hostname Router-2
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
ip address 2.2.2.2 255.255.255.0
ip pim sparse-dense-mode
!
interface Ethernet0/0
ip address 1.5.13.2 255.255.255.0
ip pim sparse-dense-mode
!
router rip
network 2.2.2.0
network 1.5.13.0
!
dial-peer voice 111 voip
destination-pattern 111
session protocol multicast
session target ipv4:237.111.0.111:22222
ip precedence 5
!
dial-peer voice 112 voip
destination-pattern 112
session protocol multicast
session target ipv4:239.194.0.10:22224
ip precedence 5
!
```

Note If you want to join the hoot and holler session directly without having to dial any session numbers, use the **connection** command and the **plar** keyword, followed by the multicast-session number.

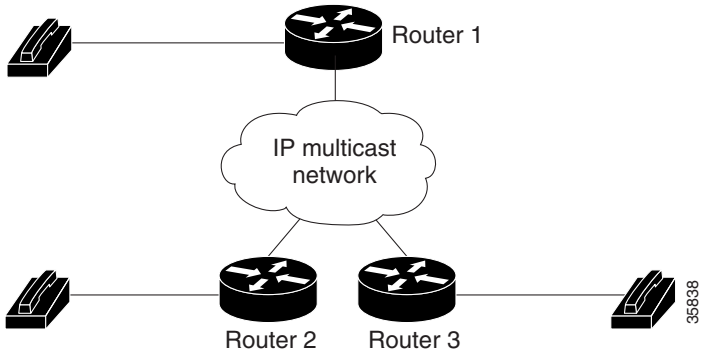
Router-3 (FXO Ports)

The following output shows that Router 4 has been configured for FXO ports:

```
hostname Router-3
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
ip address 3.3.3.3 255.255.255.0
ip pim sparse-dense-mode
!
interface Ethernet0/0
ip address 1.5.13.3 255.255.255.0
ip pim sparse-dense-mode
!
router rip
network 3.3.3.0
network 1.5.13.0
!
dial-peer voice 111 voip
destination-pattern 111
session protocol multicast
session target ipv4:237.111.0.111:22222
ip precedence 5
!
dial-peer voice 112 voip
destination-pattern 112
session protocol multicast
session target ipv4:239.194.0.10:22224
ip precedence 5
!
```

Cisco Hoot and Holler over IP with Frame-Relay Topology (One Hoot Group)

Figure 153 Cisco Hoot and Holler over IP with Frame Relay Topology



In this topology, three routers are connected using 64K Frame Relay PVCs in a hub-and-spoke topology, with Router 1 being the hub. One hoot and holler group has been defined. All three routers have been configured to traffic-shape their data and voice on the WAN to CIR, and all three routers are using IP RTP priority to guarantee QoS for the Cisco hoot and holler over IP packets. In addition, the Frame Relay broadcast-queue is disabled on the serial interfaces. This occurs because, by default, the broadcast queue is only 40 packets deep and Cisco hoot and holler over IP transmits packets at 50 packets per second. Unless the queue is disabled, some packets would be dropped and voice QoS would be degraded.

Router-1

The following output shows that Router 1 has been configured for Cisco hoot and holler over IP with Frame Relay topology:

```
hostname Router-1
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
ip address 1.1.1.1 255.255.255.0
ip pim sparse-dense-mode
!
router rip
network 1.1.1.0
network 5.5.5.0
network 5.5.6.0
!
interface Serial0/0
 no ip address
 ip pim sparse-dense-mode
 encapsulation frame-relay
 frame-relay traffic-shaping
 no frame-relay broadcast-queue
!
interface Serial0/0.1 point-to-point
 ip address 5.5.5.1 255.255.255.0
 ip pim sparse-dense-mode
 frame-relay class hoot-n-holler
```

```
 frame-relay interface-dlci 100
 frame-relay ip rtp header-compression
!
interface Serial0/0.2 point-to-point
 ip address 5.5.6.1 255.255.255.0
 ip pim sparse-dense-mode
 frame-relay class hoot-n-holler
 frame-relay interface-dlci 101
 frame-relay ip rtp header-compression
!
map-class frame-relay hoot-n-holler
 frame-relay cir 128000
 frame-relay bc 1280
 frame-relay mincir 128000
 frame-relay fragment 160
 frame-relay ip rtp priority 16384 16384 128
 no frame-relay adaptive-shaping
!
voice-port 1/0/0
voice-class permanent 1
connection trunk 111
music-threshold -30
operation 4-wire
!
dial-peer voice 1 voip
destination-pattern 111
session protocol multicast
session target ipv4:237.111.0.0:22222
ip precedence 5
```

Router-2

The following output shows that Router 2 has been configured for Cisco hoot and holler over IP with a Frame Relay topology:

```
hostname Router-2
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
ip address 2.2.2.2 255.255.255.0
ip pim sparse-dense-mode
!
router rip
network 2.2.2.0
network 5.5.5.0
!
interface Serial0/0
 no ip address
 ip pim sparse-dense-mode
 encapsulation frame-relay
 frame-relay traffic-shaping
 no frame-relay broadcast-queue
!
interface Serial0/0.1 point-to-point
 ip address 5.5.5.2 255.255.255.0
 ip pim sparse-dense-mode
 frame-relay class hoot-n-holler
```

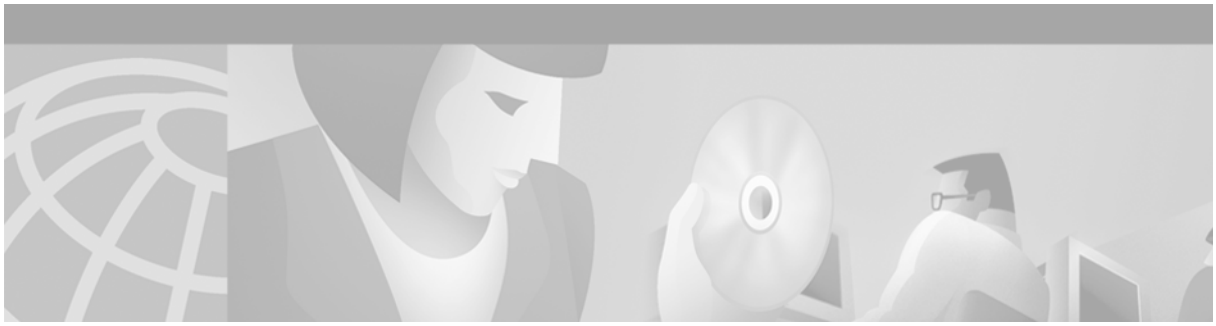
```
 frame-relay interface-dlci 100
 frame-relay ip rtp header-compression
!
map-class frame-relay hoot-n-holler
 frame-relay cir 128000
 frame-relay bc 1280
 frame-relay mincir 128000
 frame-relay fragment 160
 frame-relay ip rtp priority 16384 16383 128
 no frame-relay adaptive-shaping
!
voice-port 1/0/0
voice-class permanent 1
connection trunk 111
music-threshold -30
operation 4-wire
!
dial-peer voice 1 voip
destination-pattern 111
session protocol multicast
session target ipv4:237.111.0.0:22222
ip precedence 5
```

Router-3

The following output shows that Router 3 has been configured for Cisco hoot and holler with a Frame Relay topology:

```
hostname Router-3
!
ip multicast-routing
!
voice class permanent 1
signal timing oos timeout disabled
signal keepalive 65535
!
interface Vif1
ip address 3.3.3.3 255.255.255.0
ip pim sparse-dense-mode
!
router rip
network 3.3.3.0
network 5.5.6.0
!
interface Serial0/0
 no ip address
 ip pim sparse-dense-mode
 encapsulation frame-relay
 frame-relay traffic-shaping
 no frame-relay broadcast queue
!
interface Serial0/0.1 point-to-point
 ip address 5.5.6.2 255.255.255.0
 ip pim sparse-dense-mode
 frame-relay class hoot-n-holler
 frame-relay interface-dlci 101
 frame-relay ip rtp header-compression
!
map-class frame-relay hoot-n-holler
 frame-relay cir 128000
 frame-relay bc 1280
 frame-relay mincir 128000
```

```
 frame-relay fragment 160
 frame-relay ip rtp priority 16384 16383 128
 no frame-relay adaptive-shaping
!
voice-port 1/0/0
voice-class permanent 1
connection trunk 111
music-threshold -30
operation 4-wire
!
dial-peer voice 1 voip
destination-pattern 111
session protocol multicast
session target ipv4:237.111.0.0:22222
ip precedence 5
```



Enhanced Voice Services for Japan for Cisco 800 Series Routers

This appendix describes the Enhanced Voice Services for Japan Cisco IOS features, including INS-NET-64 voice features.

This appendix includes the following sections:

For a complete description of the commands used in this appendix, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this appendix, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this appendix, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Enhanced Voice Services Overview

The Enhanced Voice Services for Japan Cisco IOS features consist of the following voice capabilities for the Cisco 800 series routers:

- Caller ID—Provides analog caller ID support for Japanese-language display, Caller ID- equipped, analog telephones. The Cisco 800 series router receives the caller ID information from the INS-NET-64 switch. The router software prepares the caller ID with a tone, transmits the caller ID to plain old telephone service (POTS) port 1 or 2 on the router, and displays the caller ID on the telephone.
- Call Blocking on Caller ID—Allows Cisco 800 series routers to reject an incoming voice call, based on local directory number (LDN) caller IDs. Using the command-line interface (CLI), you can configure blocking for up to ten caller ID numbers for each LDN.

- Local Call Waiting—Notifies you with a call-waiting tone of an incoming call while you are already connected to a telephone call. You can place the first call on hold by pressing the on-and-off-hook button (flash), connect to the second call, and then return to the first call after finishing with the second.

  The feature uses both B channels of the ISDN line, enabling local call-waiting support on the router. Unlike standard ISDN call waiting, local call waiting does not require a subscription to call waiting from a service provider.

- E Ya Yo—Conceals the caller ID of the outgoing call from the receiving device. To activate the feature, dial 184 before dialing the number of the receiving device, as specified in the Nippon Telegraph and Telephone (NTT) Communications Corporation user manual. This feature is specific to NTT Communications Corporation switches and is offered free of charge. The router handles this feature as a regular outgoing call and requires no special operation.

- Voice Warp—On the INS-NET-64 switch, forwards all incoming calls for a terminal device to another device. Voice-warp registration, activation, and deactivation requests are sent to the switch for each LDN. The routers support the registration, activation, and deactivation requests for devices attached to the PHONE 1 or 2 port. The forwarding function itself is performed by the INS-NET-64 switch. This feature can be deactivated after its registration and activation phases.

  During the registration phase of the device, you can:

  - Create a list of forwarding destination numbers and select one as the active destination.
  - Specify whether an announcement is made to the caller, to the forwarding device, or both, when the call is forwarded.
  - Set the no-answer timer parameter from 5 to 60 seconds at 5-second intervals. This setting affects the redirection of calls when the voice-warp feature is activated.

  During the activation phase, you determine whether calls are redirected all the time or only if the receiving device is busy or does not answer within the specified no-answer time period. You can use the telephone keypad dialing sequence as specified in the NTT user manual for any of the operations described above and to hear the Voice Warp registration details for a local device.

- Voice Select Warp—An enhanced version of the Voice Warp feature. You can create a list of incoming caller IDs that is used in call redirection, either by redirecting incoming calls only from matching caller IDs, or by redirecting all calls except those from matching caller IDs. You can use the telephone keypad dialing sequence as specified in the NTT user manual for any of the Voice Select Warp feature operations and to hear the Voice Warp registration details for a local device.

- Nariwake—Checks for caller IDs that you register (using the telephone keypad for each LDN) and presents a distinctive ring to the telephone port receiving the incoming call if a match is detected. The routers provide three different ring cadences that you can set for calls from both registered and unregistered callers.

  The default ring cadence setting is ring 1 for registered callers and ring 0 for unregistered callers. The on-and-off period for normal ringing signals (ring 0) and ringing signals for Nariwake service (ring 1) are defined in the NTT user manual.

  The number of caller IDs you can register for each LDN at one time is defined by the INS-NET-64 switch, not by the router. You can register this feature with the list of caller IDs for each LDN, cancel the registration for the LDN, or get registration information from the INS-NET-64 switch. You can use the telephone keypad dialing sequence as specified in the NTT Communications Corporation user manual for any of the Nariwake feature operations and to hear the Nariwake registration details for a local device.

- Trouble Call Blocking—Also described as *nuisance telephone call refusal service* by INS Net. The network rejects all incoming calls to a particular telephone number from a troublesome caller. You do not have to specify the actual telephone number of the caller.

  When activated, the caller hears a standard telephone announcement and a disconnect message. For information about the announcement or message, see the NTT user manual.

  You are not automatically notified of incoming call attempts. However, to confirm call blocking results, you can listen to an announcement listing the number of incoming calls from blocked telephone numbers during the previous two months.

  The number of callers that you can block is defined by the service provider at the time the service is activated. If you request an additional telephone number to block beyond the defined limit, the oldest number is discarded (unblocked) before the new number is registered.

  To add a new number, you must hang up the telephone, go off-the-hook, and dial the call-blocking telephone keypad sequence within 60 seconds. When the feature is activated, you receive a recorded announcement indicating whether or not the activation is successful.

  The feature can be turned off for either the last added (blocked) number or for all call-blocked numbers. A recorded announcement indicates the changes after they are made.
- I Number—Supports the use of multiple terminal devices with one subscriber line. The telephone numbers of the subscriber line and router ports are assigned by the service provider. Calls coming into any of the assigned numbers are routed through the same subscriber line to the terminal device attached to the target port.
- POTS Dial—Supports the POTS dial feature for Japanese telephones. Using a dial application on your workstation, you can dial a telephone number for the POTS port on the router.

  If the telephone is on the hook when you issue the dial command, the router rings the telephone, waits until the telephone is taken off the hook, and then dials the requested number. If the telephone is off the hook when you issue the command, the router dials the requested number.
- POTS Disconnect—Disconnects a telephone number from the POTS port on the router.

The Enhanced Voice Services for Japan Cisco IOS features provide the following benefits:

- Caller ID—Provide analog caller ID support for Japanese caller ID-equipped telephones.
- Call Blocking on Caller ID—Reject incoming voice calls based on LDN caller IDs.
- Local Call Waiting—Provide call waiting on a local basis for Cisco 800 series routers.
- E Ya Yo—Prevent the caller ID of an outgoing call from being visible to a receiving device.
- Voice Warp—On a switch, forward registered incoming calls for a terminal device to another terminal device. List more than one forwarding destination number in the switch register and then select one to be the active number. Specify an announcement to be heard on the caller side, the forwarding side, or both when a call is forwarded.
- Voice Select Warp—Create a registration list of caller IDs, and use it to redirect incoming calls. Choose to ignore the registration list, which causes functionality to be the same as Voice Warp.
- Nariwake—Provide distinctive ring cadences for registered caller IDs to telephone ports receiving incoming calls.
- Trouble Call Blocking—Refuse nuisance telephone calls.
- I Number—Use one subscriber line for multiple terminal devices.
- POTS Dial—Dial a telephone number on a Cisco 800 series router POTS port by using a dial application on your workstation.
- POTS Disconnect—Disconnect a telephone number from a Cisco 800 series router POTS port.

Enhanced Voice Services Limitations

The Enhanced Voice Services for Japan Cisco IOS features have the following restrictions:

- You must subscribe to the NTT services to use the Enhanced Voice Services for Japan Cisco IOS features. Therefore, except for the Call Blocking on Caller ID feature and Local Call Waiting, support is limited to telephone service inside Japan.
- Caller ID
  - You must subscribe to caller ID service before using this feature.
  - In Japan, the analog caller ID feature supports only Japanese caller ID telephones.
- Call Blocking on Caller ID
  - You must subscribe to caller ID service before using this feature.
  - This function is not enabled during setup; it is only enabled if you enter caller ID numbers for blocking through the CLI.
  - The routers store a maximum of ten caller ID telephone numbers to block. Cisco 800 series routers do not accept additional caller ID numbers if ten numbers already exist. In this case, you must remove a number before adding another caller ID number for blocking.
- Local Call Waiting
  - This feature is not supported if any of the interactive voice response (IVR) features (such as voice warp, voice select warp, and Nariwake) are in use.
  - The call waiting feature is provided locally; therefore each call must have its own separate B channel. Local Call Waiting is not available if data traffic is already on-going or if both B channels are in use, for example, if POTS 1 and POTS 2 are already connected.
  - If an ISDN line already supports Call Waiting before Local Call Waiting is configured on a Cisco 811 or 813 router, the router activates ISDN Call Waiting instead of Local Call Waiting.
- Voice Warp
  - You must subscribe to the Voice Warp service before using this feature.
  - Activating the Voice Warp feature disables support for the Call Waiting feature for both local and network calls.
  - Status information for this feature is delivered over voice only.
  - The routers support this feature for one only LDN. If more than one LDN is configured, only the primary LDN can be used with this feature.
- Voice Select Warp—All Voice Warp limitations apply to the Voice Select Warp feature.
- Nariwake
  - You must subscribe to Nariwake service before using this feature.
  - Activating the Nariwake feature disables support for the Call Waiting feature for both local and network calls.
  - The Cisco 800 series routers support this feature for one LDN only. If more than one LDN is configured, only the primary LDN can be used with this feature.

- Trouble Call Blocking
  - The maximum number of troublesome callers you can block is defined when the service is activated. If you request to block more than the maximum number, the oldest blocked number must be unblocked before the new telephone number can be registered.
  - When multiple NTT services are provided with the troublesome call refusing feature, the features could possibly limit or interact with each other.
  - The Cisco 800 series routers support this feature for one LDN only. If more than one LDN is configured, only the primary LDN can be used with this feature.
- I Number—You must subscribe to the I Number service before using this feature.

Related Documents for Enhanced Voice Services

The following documents provide additional platform-specific information:

- Cisco 800 Series Routers
  - *Cisco 800 Series Router Quick Start Guide*
  - *Cisco 800 Series Routers Hardware Installation Guide*
  - *Cisco 800 Series Routers Software Configuration Guide*
- Cisco 805 Routers
  - *Cisco 805 Router Hardware Installation Guide*
  - *Quick Start Guide – Setting Up the Cisco 805 Router*
  - *Cisco 805 Router Software Configuration Guide*
- Cisco 811 and 813 Routers
  - *Cisco 811 and Cisco 813 Routers Hardware Installation Guide*
  - *Quick Start Guide: Setting Up Cisco 811 and Cisco 813 Routers*

Enhanced Voice Services Prerequisite Tasks

Before using the Enhanced Voice Services for Japan Cisco IOS features, use the Cisco IOS command **pots country *jp*** to configure the router telephone ports to Japanese standards. The following requirements must also be met:

- E Ya Yo—You must subscribe to the NTT Communications Corporation E Ya Yo service and connect the router to a Japanese INS-NET-64 switch.
- Voice Warp—You must subscribe to the NTT Communications Corporation Voice Warp and Caller ID services and connect the router to a Japanese INS-NET-64 switch.
- Voice Select Warp—You must subscribe to the NTT Communications Corporation Voice Select Warp service and connect the router to a Japanese INS-NET-64 switch.
- Nariwake—You must subscribe to the NTT Communications Corporation service for distinctive incoming calls (distinctive ringing).
- Trouble Call Blocking—You must subscribe to the NTT Communications Corporation service for refusing troublesome calls.

Enhanced Voice Services Configuration Task List

Many of the Enhanced Voice Services for Japan Cisco IOS features were developed for other Cisco routers before they were ported to Cisco 800 series routers. In some cases, CLI commands were created or modified to allow the features to run on Cisco 800 series routers. The following sections provide step-by-step instructions for configuring only those features that require new or changed Cisco IOS commands specifically created or modified to run on Cisco 800 series routers.

The Local Call Waiting feature is enabled by a single command in global configuration mode; for command syntax and examples, see the **pots call-waiting** command in the *Cisco IOS Voice, Video, and Fax Command Reference*. The features POTS Dial and POTS Disconnect are also single commands in EXEC mode; for command syntax and examples, see the **test pots dial** and **test pots disconnect** commands in the *Cisco IOS Voice, Video, and Fax Command Reference*.

See the following sections for configuration tasks for Enhanced Voice Services for Japan:

- Configuring Caller ID, page 902 (optional)
- Configuring Call Blocking on Caller ID, page 902 (optional)
- Configuring Nariwake, page 903 (optional)
- Configuring I Number, page 903 (optional)
- Monitoring and Maintaining Enhanced Voice Services, page 904 (optional)

Configuring Caller ID

To configure Caller ID, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# pots country jp`	Configures the router telephone ports to Japanese standards.
Step 2	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode, and selects the POTS port.
Step 3	`Router(config-dial-peer)# caller-id`	Enables the Caller ID feature.

Configuring Call Blocking on Caller ID

To configure Call Blocking on Caller ID, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode, and selects the POTS port.
Step 2	`Router(config-dial-peer)# block-caller` *number*	Blocks the caller ID: *number*. For example, blocks incoming calls from the telephone number 408-555-1234.

Configuring Nariwake

To configure Nariwake, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# pots country jp`	Configures the router telephone ports to Japanese standards.
Step 2	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode, and selects the POTS port.
Step 3	`Router(config-dial-peer)# registered-caller ring` *cadence*	Configures the Nariwake service registered caller ring cadence. For example, you could set the ring cadence for registered callers to 2.
Step 4	`Router(config-dial-peer)# destination-pattern not-provided`	(Optional) If you also subscribe to I Number and Dial-In services, configures a dial peer.

Configuring I Number

To configure I Number, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri` *number*	Enters interface configuration mode and specifies the basic rate interface number, such as **bri0**.
Step 2	`Router(config-if)# isdn i-number` *n1 ldn1*	Configures the first router POTS port to use a single subscriber line, such as 5551234.
Step 3	`Router(config-if)# isdn i-number` *n2 ldn2*	Configures the second router POTS port to use a single subscriber line, such as 5556789.
Step 4	`Router(config-if)# exit`	Exits interface configuration mode for the basic rate interface configuration.
Step 5	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode, and selects the POTS port.
Step 6	`Router(config-dial-peer)# destination-pattern 5551234`	Sets the first dial-peer destination pattern to the corresponding LDN, such as 5551234.
Step 7	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode for the first **destination-pattern** configuration.
Step 8	`Router(config)# dial-peer voice` *number* `pots`	Enters dial-peer configuration mode, and selects the POTS port.
Step 9	`Router(config-dial-peer)# destination-pattern` *number*	Sets the second dial-peer destination pattern to the corresponding LDN.
Step 10	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode for the second **destination-pattern** configuration.

Monitoring and Maintaining Enhanced Voice Services

To monitor your enhanced voice services configuration, use the following command:

Command	Purpose
router# **show pots csm** *port*	Displays information about the current state of calls and the most recent event received by the call switching module (CSM) in the router.

Enhanced Voice Services Configuration Examples

This section contains the following examples:

Caller ID Example

The following example shows how to configure a router to use the Caller ID feature.

```
dial-peer voice 1 pots
 caller-id
```

Call Blocking on Caller ID Example

The following example shows how to configure a router to use the Call Blocking on Caller ID feature. This example configures a router to block calls from a caller whose Caller ID number is 408-555-1234.

```
dial-peer voice 1 pots
 block-caller 4085551234
```

Local Call Waiting Example

The following example shows how to enable the Local Call Waiting feature on a router.

```
pots call-waiting local
```

Nariwake Example

The following two examples show how to configure a router to use the Nariwake feature.

The first example sets the ring cadence for registered callers to 2:

```
pots country jp
dial-peer voice 1 pots
 registered-caller ring 2
```

The second example adds the **destination-pattern not-provided** command, which is needed if you also subscribe to the I Number and dial-in services:

```
pots country jp
dial-peer voice 1 pots
 registered-caller ring 2
 destination-pattern not-provided
```

I Number Example

The following example shows how to configure a router to use the ISDN I Number feature so that several terminal devices can share one subscriber line. This example shows the configuration commands for two LDNs configured under interface BRI0:

```
interface bri0
 isdn i-number 1 5551234
 isdn i-number 2 5556789
 exit
dial-peer voice 1 pots
 destination-pattern 5551234
 exit
dial-peer voice 2 pots
 destination-pattern 5556789
 exit
```

POTS Dial Example

The following example shows how to use the POTS Dial feature.

The POTS dial command shown below dials the telephone number 408-555-1234:

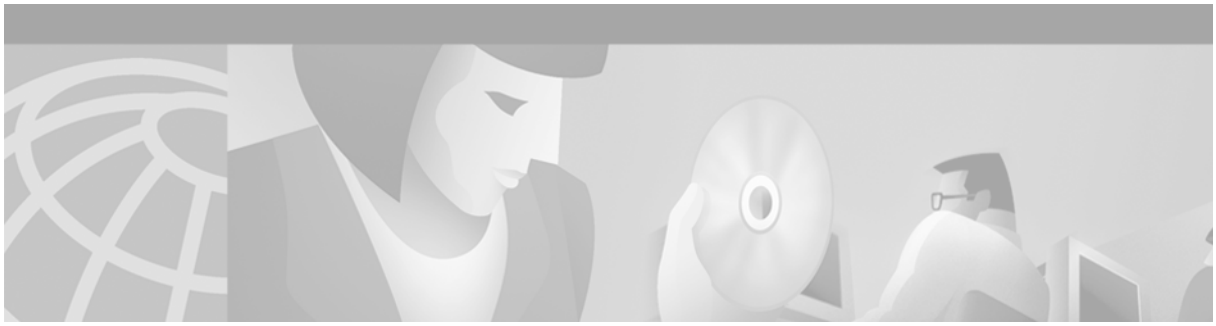
```
test pots 1 dial 4085551234#
```

POTS Disconnect Example

The following example shows how to use the POTS Disconnect feature.

The POTS disconnect command shown below disconnects a telephone call from POTS port 1:

```
test pots 1 disconnect
```



Managing Cisco AS5300 Voice Feature Cards

This appendix explains how to manage voice feature cards (VFCs) for the Cisco AS5300 and contains the following sections:

For a complete description of the commands used in this appendix, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this appendix, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this appendix, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

VFC Management Overview

Voice feaure cards (VFCs) for the Cisco AS5300 come with a single bundled image of VCWare stored in VFC Flash memory.

DSPWare is stored as a compressed file within VCWare; you must unbundle VCWare to install DSPWare into Flash memory. During the unbundling process, two default lists (the default file list and the capability list) are automatically created, populated with default files from that version of VCWare, and stored in VFC Flash memory. The default file list contains the filenames indicating which files are initially loaded into DSP upon bootup. The capability list defines the set of codecs that can be negotiated for a voice call.

Table 62 shows the extension types defined for these embedded firmware files.

Table 62 VFC Firmware Extensions

Firmware	Filenames	Description
VCWare	vcw-vfc-*	Latest version of VCWare stores in Flash memory, including: • Datapath engine • Message dispatcher • DSP manager • VC manager • Process scheduler
DSPWare	btl-vfc-*	DSP bootloader
	cor-vfc-*	Core operating system and initialization
	bas-vfc-*	Base voice
	cdc-*-*	Voice codec files
	fax-vfc-*	Fax relay files

VFC Management Task List

VFC management enables you to add versions of VCWare to Flash memory (download and unbundle files), erase files contained in Flash memory, add files to the default file list and capability list, and delete files from the default file lists and capability lists. These tasks are described in the following sections:

- Downloading VCWare, page 908
- Copying Flash Files to the VFC, page 910
- Unbundling VCWare, page 911
- Adding Files to the Default File List, page 912
- Adding Codecs to the Capability List, page 912
- Deleting Files from VFC Flash Memory, page 913
- Erasing the VFC Flash Memory, page 913

Downloading VCWare

To download software to your VFC, perform the following tasks:

- Determine that the version of VFC ROM Monitor software is compatible with your installed Cisco IOS image. VFC ROM version 1.2 requires Cisco IOS image 0.14.1 (1.6 NA1) or later. VFC ROM Monitor version 1.2 can be made to work with Cisco IOS image 0.13 (or later) by appending the suffix ".VCW" to the VCWare image stored in VFC Flash memory.
- Determine whether the VFC is in VCWare mode or ROM Monitor mode. The mode, whether VCWare or ROM Monitor, determines which procedure you will use to download software to the VFC.
- Download the software using the appropriate procedure.

Note For each installed VFC, perform the tasks described in the following sections to upgrade system software on that VFC.

Identifying the VFC Mode

To identify the mode (whether VCWare or ROM Monitor), use the following command in privileged EXEC mode:

Command	Purpose
`Router# show vfc slot board`	Determines whether your VFC is operating in VCWare mode or ROM Monitor mode.

If the mode is VCWare, the VFC status will be "VCWARE running." If the mode is ROM Monitor, the VFC status will be "ROMMON."

Downloading Software (VCWare Mode)

To download VFC software to the VFC while the VFC is in VCWare mode, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# erase vfc slot`	Erases the Flash memory.
Step 2	`Router# show vfc slot directory`	Verifies that the VFC Flash memory is indeed empty.

Step 3	**Downloading from the Router Motherboard**
	`Router# copy flash: vfc:`
	or
	Downloading from a TFTP Server
	`Router# copy tftp: vfc:`
		Downloads the VCWare file from the router motherboard into the Flash memory on the VFC.
		
		Downloads the VCWare file from a TFTP server into the Flash memory on the VFC.
		Note The colons in this command are required.

Step 4	`Router# clear vfc slot`	Reboots the VFC.
Step 5	`Router# show vfc slot board`	Displays whether the VFC is back up in VCWare mode.
Step 6	`Router# show vfc slot directory`	Verifies that VCWare is in the VFC flash.
Step 7	`Router# unbundle vfc slot`	Unbundles the DSPWare from the VCWare and configures the default file list and the capability list.
Step 8	`Router# show vfc slot directory`	Verifies that the DSPWare has been unbundled.
Step 9	`Router# show vfc slot default-file`	Verifies that the default file list has been populated.
Step 10	`Router# show vfc slot cap-list`	Verifies that the capability list has been populated.

Reboot the router in order for these changes to take effect.

Note If the VFC ROM is version 1.1, the image name must end in ".VCW." If the VFC ROM is version 1.2, the image name must start with "vcv-."

Downloading Software (ROM Monitor Mode)

To download VFC software to the VFC while the VFC is in ROM Monitor mode, use the following commands beginning in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# clear vfc slot purge`	Erases the VFC Flash memory.

Step 2	**Downloading from the Router Motherboard**
	`Router# copy flash: vfc:`
	or
	Downloading from a TFTP Server
	`Router# copy tftp: vfc:`
		Downloads the VCWare file from the router motherboard into the Flash memory on the VFC.
		
		Downloads the VCWare file from a TFTP server into the Flash memory on the VFC.
		Note The colons in this command are required.

Step 3	`Router# clear vfc slot`	Reboots the VFC.
Step 4	`Router# show vfc slot board`	Displays whether the VFC is back up in VCWare mode.
Step 5	`Router# show vfc slot directory`	Verifies that VCWare is in the VFC flash.
Step 6	`Router# unbundle vfc slot`	Unbundles the DSPWare from the VCWare and configures the default file list and the capability list.
Step 7	`Router# show vfc slot directory`	Verifies that the DSPWare has been unbundled.
Step 8	`Router# show vfc slot default-file`	Verifies that the default file list has been populated.
Step 9	`Router# show vfc slot cap-list`	Verifies that the capability list has been populated.

Reboot the router in order for these changes to take effect.

Note The image name must start with "vcw-."

Copying Flash Files to the VFC

As mentioned, each VFC comes with a single bundled image of VCWare stored in Flash memory. VoIP for the Cisco AS5300 offers two different ways to copy new versions of VCWare to the VFC Flash memory: either by downloading the VCWare image from the router motherboard or by downloading the VCWare image from a TFTP server.

Downloading VCWare to the VFC from the Router Motherboard

To download the VCWare file from the router motherboard to VFC Flash memory, use the following command in privileged EXEC mode:

Command	Purpose
`Router# copy flash: vfc:`	Downloads (copies) the VCWare file from the router motherboard into the Flash memory on the VFC. **Note** The colons in this command are required.

Downloading VCWare to the VFC from a TFTP Server

To download the latest version of VCWare from a TFTP server, first ensure that the file is stored on the TFTP server. If you have a copy of the current version of VCWare on disk, you must store that image on a TFTP server before you can download the file to the VFC Flash memory.

To copy the flash file from a TFTP server, use the following command in privileged EXEC mode:

Command	Purpose
`Router# copy tftp: vfc:`	Downloads (copies) the VCWare file from a TFTP server into the Flash memory on the VFC. **Note** The colons in this command are required.

Unbundling VCWare

In order for the DSPWare to be loaded into the Flash memory on a VFC, the VCWare needs to be unbundled, and the two necessary default lists (default file list and capability list) need to be created and populated with the appropriate default files for that version of DSPWare. Table 63 shows the files associated with each firmware file.

Table 63 VFC Firmware Filenames

Firmware	Filenames
VCWare	vcw-vfc-mz.0.15.bin
DSPWare Initialization and Static Files	btl-vfc-1.0.14.0.bin cor-vfc-1.0.14.0.bin jbc-vfc-1.0.14.0.bin
DSPWare Overlay Files	bas-vfc-1.0.14.0.bin cdc-g711-1.0.14.0.bin cdc-g729-1.0.14.0.bin fax-vfc-1.0.14.0.bin

To unbundle the current running image of VCWare, use the following command in privileged EXEC mode:

Command	Purpose
`Router# unbundle vfc slot`	Unbundles the current image of VCWare. This action unbundles the DSPWare from the VCWare and configures the default file list and the capability list.

Adding Files to the Default File List

When you unbundle VCWare, the default file list is automatically created and populated with the default files for that version of VCWare. The default file list indicates which files will be initially loaded into DSP when the router boots up. The following example shows the output from the **show vfc default-file** command, which displays the contents of the default file list:

```
router# show vfc 1 default-file

Default List for VFC in slot 1:
1. btl-vfc-1.0.13.0.bin
2. cor-vfc-1.0.1.bin
3. bas-vfc-1.0.1.bin
4. cdc-g729-1.0.1.bin
5. fax-vfc-1.0.1.bin
6. jbc-vfc-1.0.13.0.bin
```

Under most circumstances, these default files should be sufficient. If you need to, you can add a file (from those stored in VFC Flash memory) to the default file list or replace an existing file from the default file list. When you add a specific file to the default file list, it replaces the existing default for that extension type.

To select a file to be added to the default file list, use the following command in global configuration mode:

Command	Purpose
`Router(config)# default-file vfc`	Selects a file stored in the Flash memory to be added to the default file list.

Adding Codecs to the Capability List

The capability list defines the set of codecs that can be negotiated for a voice call. Like the default file list, the capability list is created and populated when VCWare is unbundled and DSPWare added to VFC Flash memory. The following example shows the output from the **show vfc cap-list** command, which displays the contents of the capability list:

```
router# show vfc 1 cap-list

Capability List for VFC in slot 1:
1. fax-vfc-1.0.1.bin
2. bas-vfc-1.0.1.bin
3. cdc-g729-1.0.1.bin
4. cdc-g711-1.0.1.bin
5. cdc-g726-1.0.1.bin
```

```
6. cdc-g728-1.0.1.bin
7. cdc-gsmfr-1.0.1.bin
```

VFC management lets you add codec files to the capability list to meet the needs of your specific telephony network.

Note The capability list does not indicate codec preference; it simply reports the codecs that are available. The session application decides which codec to use.

To add a codec overlay file to the capability list, use the following command in global configuration mode:

Command	Purpose
`Router(config)# cap-list` *file-name* `vfc` *slot-number*	Selects a codec overlay file to be added to the capability list.

Deleting Files from VFC Flash Memory

In some instances, you might need to delete a file from the default file list or the capability list, or you might need to revert to a previous version of VCWare stored in Flash memory. To delete a file, you must identify and delete the file from VFC Flash memory. Deleting a file from Flash memory removes the file from the default file list and from the capability list (if the deleted file is included on those lists).

To delete a file from VFC Flash memory, use the following command in privileged EXEC mode:

Command	Purpose
`Router# delete` *file-name* `vfc` *slot*	Deletes the specified file from the Flash memory on the VFC.

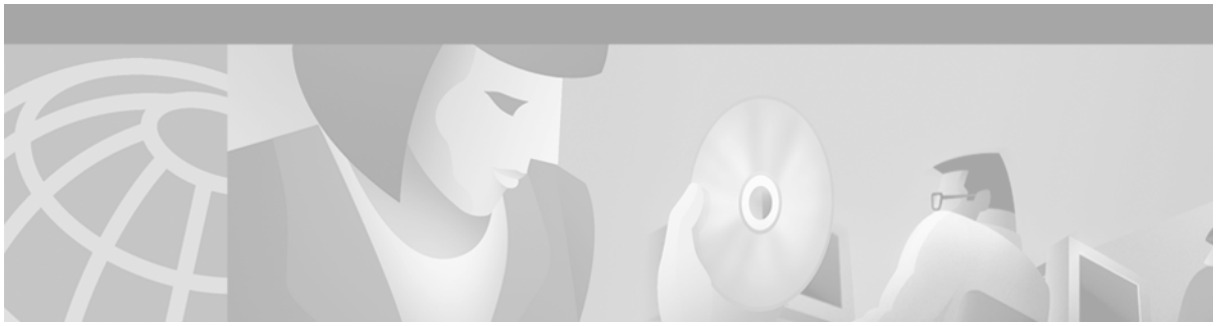
Erasing the VFC Flash Memory

When you upgrade to a later version of VCWare, the new files are stored in VFC Flash memory, along with the existing files already stored in VFC Flash memory—the new files do not overwrite existing files. Consequently, you will eventually need to erase the contents of VFC Flash memory to free VFC Flash memory space. Erasing VFC Flash memory removes the entire contents stored in Flash memory, including the default file list and the capability list.

To erase the Flash memory on a specific VFC, use the following command in privileged EXEC mode:

Command	Purpose
`Router# erase vfc` *slot*	Erases the Flash memory on the VFC. **Caution** This command removes *all* files stored in the Flash memory of the VFC in the specified slot.

Note For more information about VFC management commands, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*.



Global System for Mobile Communications Full Rate and Enhanced Full Rate Codecs

This appendix describes the global system for mobile communications (GSM) full rate (FR) and enhanced full rate (EFR) codecs feature. The appendix includes the following sections:

For a complete description of the commands used to configure VoIP for modem support, refer to the *Cisco IOS Voice, Video, and Fax Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature in this chapter, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms" section in the "Using Cisco IOS Software" chapter.

Global System for Mobile Communications Full Rate and Enhanced Full Rate Codecs Overview

The global system for mobile communications full rate and enhanced full rate codecs supports Cisco mobile office network (MNET) GSM mobile telephony products and solutions. By leveraging the IP functionality of the Cisco network and its voice gateways, these products and solutions enhance the effectiveness of individuals in an enterprise environment. The feature includes GSM full rate and enhanced full rate codecs in the digital signal processor (DSP) firmware of the voice gateway and supplementary services, such as blind call transfer.

Call transfer allows an H.323 endpoint to redirect an answered call to another H.323 endpoint. Cisco gateways support H.450.2 call transfer as the transferred and transferred-to party. The transferring endpoint must be an H.450-capable terminal; the Cisco gateway cannot act as the transferring endpoint. Gatekeeper-controlled or gatekeeper-initiated call transfer is not supported.

The global system for mobile communications full rate and enhanced full rate codecs is supported on the following platforms:

- Cisco VG200
- Cisco 2600, 3600, 7200, and 7500 series routers
- Cisco AS5300 universal access server

The Cisco voice gateway supports the Cisco MNET solution.

Prerequisite Tasks and Restrictions

Before configuring your Cisco AS5300 to use Voice over IP (VoIP), refer to *Cisco AS5300 Voice-over-IP Feature Card Installation and Configuration*.

The following restrictions apply to the global system for mobile communications full rate and enhanced full rate codecs:

- Call manager and IP phones are not integrated into the MNET solution. The endpoints that can interwork with the user are internal and external interfaces connected through an H.323 gateway, such as PBX users, Foreign Exchange Station (FXS) and Foreign Exchange Office (FXO) analog interfaces, and T1 channel-associated signaling (CAS) and T1 primary rate interface (PRI) digital interfaces.
- For call transfer, only blind transfer is supported.
- Call diversion according to H.450.3 is not supported.
- GSM codec is converted to pulse code modulation (PCM) via the voice gateway. Transcoding of GSM to another code type is not supported.
- The Cisco GSM mobility controller provides centralized dialing plan management and routing but does not provide RAS (registration, admission, and status) according to H.323 standards.

GSM Configuration Tasks

See the following section to configure the Cisco global system for mobile communications full rate and enhanced full rate codecs feature. The “Configuring Dial Peers” configuration task is required.

Configuring Dial Peers

The H.323 gateway must be configured to interwork with the Cisco GSM mobility controller as a peer-to-peer H.323 entity and must also be configured to be H.450 capable. To configure dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer voice` *tag* `voip`	Enters dial-peer configuration mode and defines a local dial peer that will connect to the Voice over IP (VoIP) network. The *tag* argument is one or more digits identifying the dial peer. Valid entries are from 1 to 2,147,483,647.
Step 2	`Router(config-dial-peer)# application session`	Enables H.450 features.

	Command	Purpose
Step 3	`Router(config-dial-peer)# destination-pattern [+] string [T]`	Specifies the E.164 address associated with this dial peer. The keywords and arguments are as follows: • **+**—(Optional) Specifies a character indicating an E.164 standard number. The plus sign (+) is not supported on the Cisco MC3810. • *string*—Indicates a series of digits that specify the E.164 or private dialing plan telephone number. Valid entries are the digits 0 through 9, the letters A through D, and the following special characters: – The asterisk (*) and pound sign (#)—Indicate the keys that appear on standard touch-tone dial pads. On the Cisco 3600 series only, these characters cannot be used as leading characters in a string (for example, *650). – Comma (,)—Inserts a pause between digits. – Period (.)—Matches any entered digit (this character is used as a wildcard). On the Cisco 3600 series, the period cannot be used as a leading character in a string (for example, .650). – Percent sign (%)—Indicates that the previous digit/pattern occurred zero or multiple times, similar to the wild card usage in the regular expression. – Plus sign (+)—Matches a sequence of one or more matches of the character/pattern. **Note** The plus sign used as part of the digit string is different from the plus sign that can be used in front of the digit string to indicate that the string is an E.164 standard number. – Circumflex (^)—Indicates a match to the beginning of the string. – Dollar sign ($)—Matches the null string at the end of the input string. – Backslash symbol (\)—Is followed by a single character matching that character or used with a single character having no other significance (matching that character). – Question mark (?)— Indicates that the previous digit occurred zero or one time. – Brackets ([])—Indicates a range of digits. A range is a sequence of characters enclosed in the brackets, and only numeric characters from “0” to “9” are allowed in the range. This is similar to a regular expression rule.

	Command	Purpose
		– Parentheses ( )—Indicates a pattern and is the same as the regular expression rule—for example, 408(555). Parentheses are used in conjunction with symbols ?, %, or +. For more information on applying wildcard symbols to destination patterns and the dial strings that result, see the "Configuring Dial Plans, Dial Peers, and Digit Manipulation" chapter in this configuration guide. • **T**—(Optional) Control character indicating that the **destination-pattern** value is a variable length dial string.
Step 4	`Router(config-dial-peer)# session target`	Identifies the session target (the IP address of the Cisco GSM mobility controller (GMC).
Step 5	`Router(config-dial-peer)# codec`	Specifies the voice coder rate of speech for the dial peer. (In this case, the **gsmr** keyword will be used to indicate that it is 13,200 bps.)
Step 6	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode.
Step 7	`Router(config)# dial-peer voice` *tag* `voip`	Enters dial-peer configuration mode and defines a local dial peer that will connect to the VoIP network. (This dial peer is different from the one in Step 1.)
Step 8	`Router(config-dial-peer)# application session`	Enables H.450 features.
Step 9	`Router(config-dial-peer)# incoming called-number` *string*	Specifies the incoming called number of a Multimedia Mail over Internet Protocol (MMoIP) or Plain Old Telephone Service (POTS) dial peer. The *string* argument specifies the incoming called telephone number. Valid entries are any series of digits that specify the E.164 telephone number.
Step 10	`Router(config-dial-peer)# codec`	Specifies the voice coder rate of speech for the dial peer. (In this case, the **gsmr** keyword will be used to indicate that it is 13,200 bps.)
Step 11	`Router(config-dial-peer)# exit`	Exits dial-peer configuration mode.
Step 12	`Router(config)# dial-peer voice` *tag* `pots`	Enters dial-peer configuration mode and configures a POTS dial peer. The *tag* value of the dial-peer voice POTS command uniquely identifies the dial peer. Valid entries are from 1 to 2147483647.
Step 13	`Router(config-dial-peer)# application session`	Enables H.450 features.
Step 14	`Router(config-dial-peer)# destination-pattern [+]` *string* `[T]`	Identifies the telephone number associated with this dial peer. For an explanation of the keywords and arguments, see Step 3 in this configuration task table.
Step 15	`Router(config-dial-peer)# port` *slot-number/subunit-number/port*	Associates this dial peer with a specific logical dial interface.

Verifying Gateway Configuration

To confirm the gateway configuration, perform the following steps:

Step 1 Enter the **show dial-peer voice** command to display codec information:

```
Router# show dial-peer voice 555

VoiceOverIpPeer555
         information type = voice,
         tag = 555, destination-pattern = `',
         answer-address = `', preference=0,
         numbering Type = `unknown'
         group = 555, Admin state is up, Operation state is up,
         incoming called-number = `4085264320', connections/maximum =
0/unlimited,
         DTMF Relay = disabled,
         modem passthrough = system,
         huntstop = disabled,
         in bound application associated:DEFAULT
         out bound application associated:
         permission :both
         incoming COR list:maximum capability
         outgoing COR list:minimum requirement
         type = voip, session-target = `',
         technology prefix:
         settle-call = disabled
         ip precedence = 0, UDP checksum = disabled,
         session-protocol = cisco, session-transport = udp, req-qos =
best-effort,
         acc-qos = best-effort,
         fax rate = voice,   payload size =  20 bytes
         fax protocol = system
         fax NSF = 0xAD0051 (default)
         codec = gsmefr,   payload size =  32 bytes,
codec display
         Expect factor = 0, Icpif = 20,
         Playout:Mode adaptive,
         Expect factor = 0,
         Max Redirects = 1, Icpif = 20,signaling-type = cas,
         CLID Restrict = disabled
         VAD = enabled, Poor QOV Trap = disabled,
         voice class perm tag = `'
         Connect Time = 0, Charged Units = 0,
         Successful Calls = 0, Failed Calls = 0,
         Accepted Calls = 0, Refused Calls = 0,
         Last Disconnect Cause is "",
         Last Disconnect Text is "",
         Last Setup Time = 0.
```

Step 2 Enter the **show running-config** command to view **voice class codec** information.

```
Router# show running-config
Building configuration...

Current configuration:
!
version 12.2
.
.
.
!
```

```
voice class codec 99
  codec preference 1 g711alaw
  codec preference 2 g723ar53
  codec preference 3 g723r53
  codec preference 4 g726r16
  codec preference 5 g726r24
  codec preference 6 g728
  codec preference 7 g729br8
  codec preference 8 gsmefr
  codec preference 9 gsmfr
!
.
.
.
```

GSM Configuration Example

This section provides a Frame Relay for voice over IP configuration example.

For Frame Relay, it is customary to configure a main interface and several subinterfaces, one subinterface per permanent virtual connection (PVC). The following example configures a Frame Relay main interface and a subinterface so that voice and data traffic can be successfully transported:

```
interface Serial0/0
 ip mtu 300
 no ip address
 encapsulation frame-relay
 no ip route-cache
 no ip mroute-cache
 fair-queue 64 256 1000
 frame-relay ip rtp header-compression
 interface Serial0/0.1 point-to-point
 ip mtu 300
 ip address 40.0.0.7 255.0.0.0
 ip rsvp bandwidth 48 48
 no ip route-cache
 no ip mroute-cache
 bandwidth 64
 traffic-shape rate 32000 4000 4000
 frame-relay interface-dlci 16
 frame-relay ip rtp header-compression
```

In this configuration example, the main interface has been configured as follows:

- Maximum transmission unit (MTU) size of IP packets is 300 bytes.
- An IP address is not associated with this serial interface. The IP address must be assigned for the subinterface.
- Encapsulation method is Frame Relay.
- Fair queueing is enabled.
- IP Real-Time Transport Protocol (RTP) header compression is enabled.

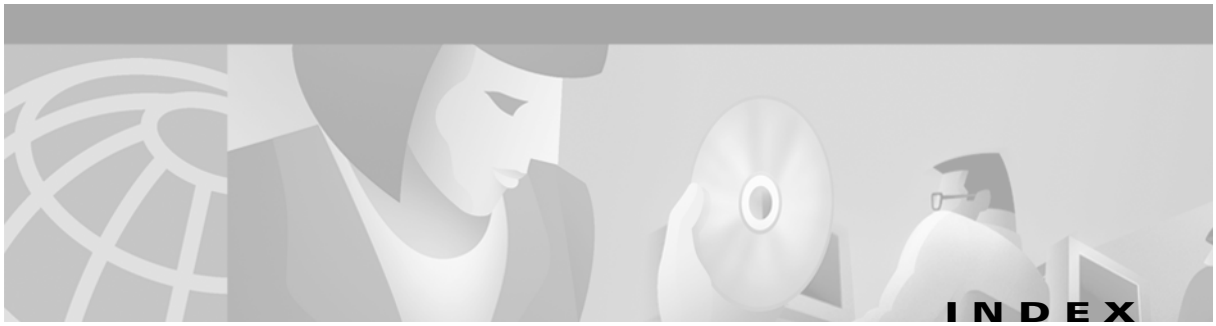
The subinterface has been configured as follows:

- MTU size is inherited from the main interface.
- IP address for the subinterface is specified.
- Bandwidth is set to 64 kbps.

- Generic traffic shaping is enabled with 32-kbps committed information rate (CIR), where Bc = 4000 bits and Be = 4000 bits.
- Frame Relay data-link connection identifier (DLCI) number is specified.
- IP RTP header compression is enabled.

Note When traffic bursts over the CIR, the output rate is held at the speed configured for the CIR (for example, traffic will not go beyond 32 kbps if the CIR is set to 32 kbps).

For more information about Frame Relay, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*.



INDEX

BC	Cisco IOS Bridging and IBM Networking Configuration Guide
DC	Cisco IOS Dial Technologies Configuration Guide
FC	Cisco IOS Configuration Fundamentals Configuration Guide
IC	Cisco IOS Interface Configuration Guide
IPC	Cisco IOS IP Routing Configuration Guide
MWC	Cisco IOS Mobile Wireless Configuration Guide
P2C	Cisco IOS AppleTalk and Novell IPX Configuration Guide
P3C	Cisco IOS Apollo Domain, Banyan VINES, DECnet, ISO CLNS, and XNS Configuration Guide
QC	Cisco IOS Quality of Service Solutions Configuration Guide
SC	Cisco IOS Security Configuration Guide
TC	Cisco IOS Terminal Services Configuration Guide
VC	Cisco IOS Voice, Video, and Fax Configuration Guide
WC	Cisco IOS Wide-Area Networking Configuration Guide
XC	Cisco IOS Switching Services Configuration Guide

Symbols

<cr> **45**

? command **44**

Numerics

23B+D **VC-27**

911 outgoing calls, MGCP support of **VC-218**

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z