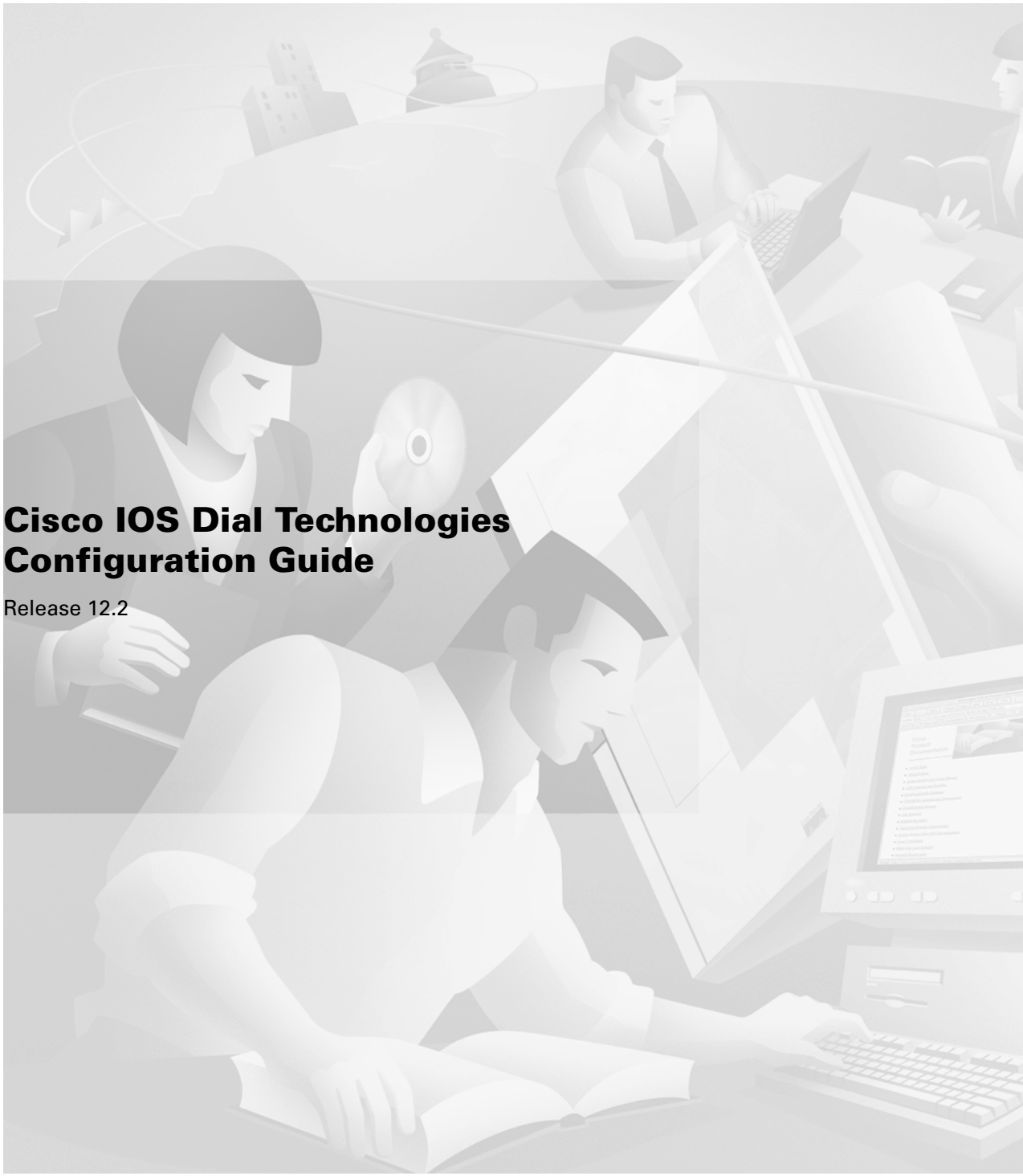
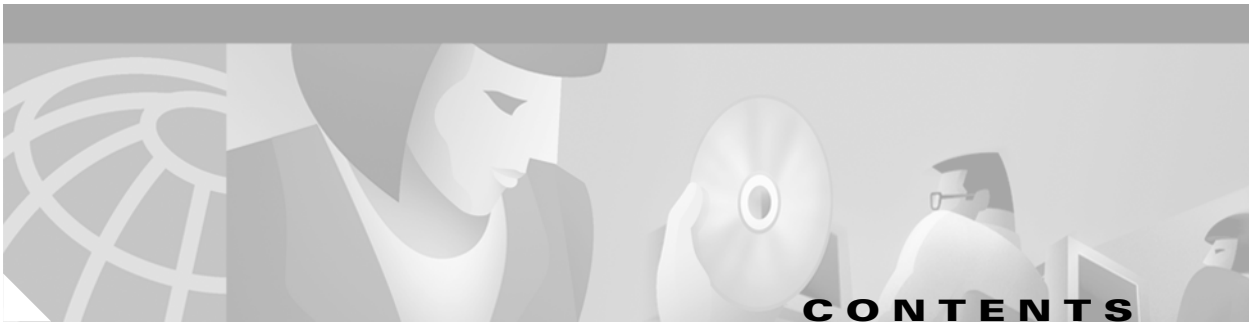


Cisco IOS Dial Technologies Configuration Guide

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-7812090=
Text Part Number: 78-12090-02



CONTENTS

DIAL INTERFACES, CONTROLLERS, AND LINES

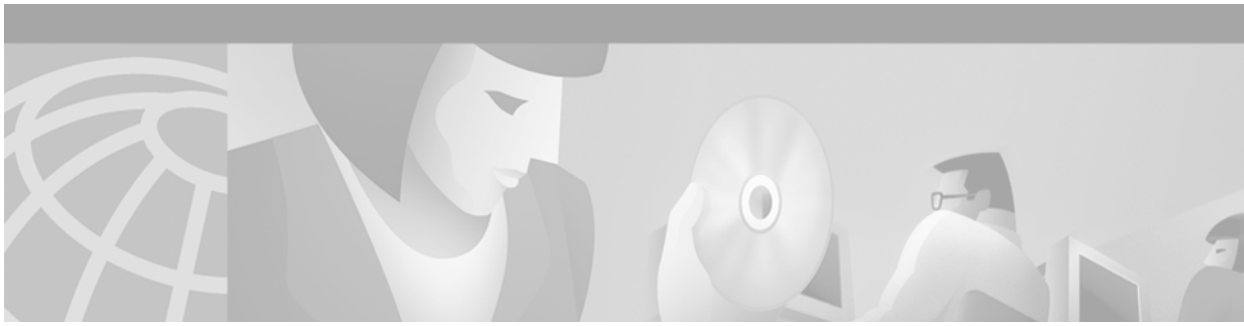
ISDN CONFIGURATION

SIGNALING CONFIGURATION

VIRTUAL TEMPLATES, PROFILES, AND NETWORKS

PPP CONFIGURATION

DIAL ACCESS SCENARIOS



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.

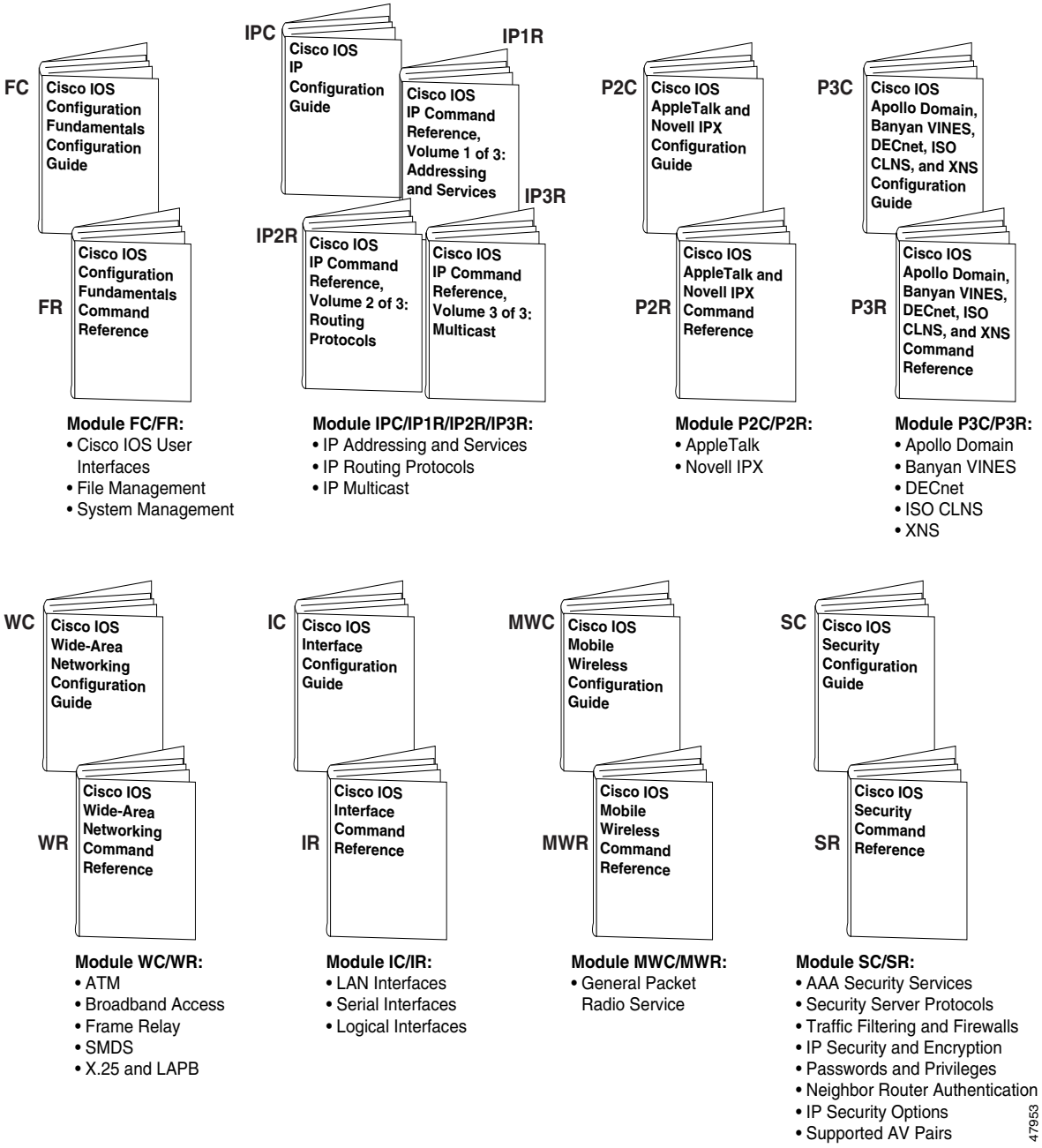
Documentation Modules

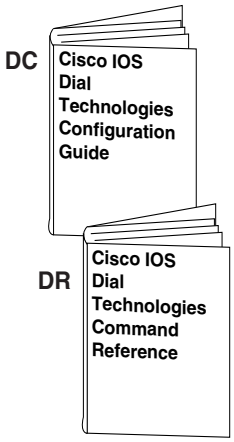
The Cisco IOS documentation modules consist of configuration guides and corresponding command reference publications. Chapters in a configuration guide describe protocols, configuration tasks, and Cisco IOS software functionality and contain comprehensive configuration examples. Chapters in a command 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.

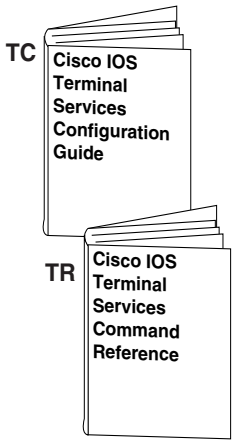
Note The abbreviations (for example, FC and FR) next to the book icons are page designators, which are defined in a key in the index of each document to help you with navigation. The bullets under each module list the major technology areas discussed in the corresponding books.

Figure 1 Cisco IOS Software Documentation Modules

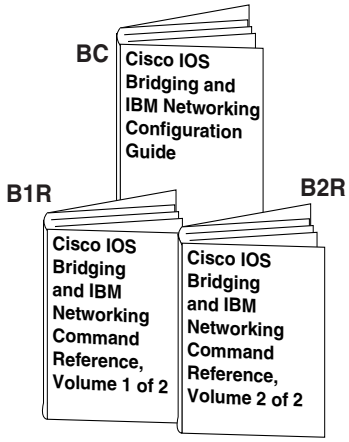




Module DC/DR:
- Preparing for Dial Access
- Modem and Dial Shelf Configuration and Management
- ISDN Configuration
- Signalling Configuration
- Dial-on-Demand Routing Configuration
- Dial-Backup Configuration
- Dial-Related Addressing Services
- Virtual Templates, Profiles, and Networks
- PPP Configuration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

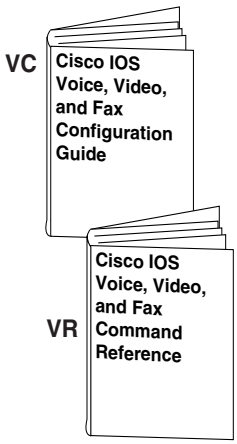


Module TC/TR:
- ARA
- LAT
- NASI
- Telnet
- TN3270
- XRemote
- X.28 PAD
- Protocol Translation

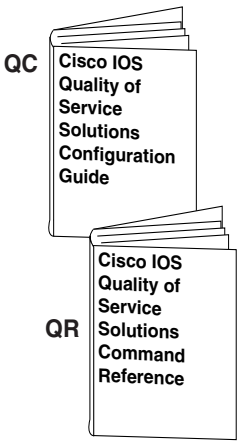


Module BC/B1R:
- Transparent Bridging
- SRB
- Token Ring Inter-Switch Link
- Token Ring Route Switch Module
- RSRB
- DLSw+
- Serial Tunnel and Block Serial Tunnel
- LLC2 and SDLC
- IBM Network Media Translation
- SNA Frame Relay Access
- NCIA Client/Server
- Airline Product Set

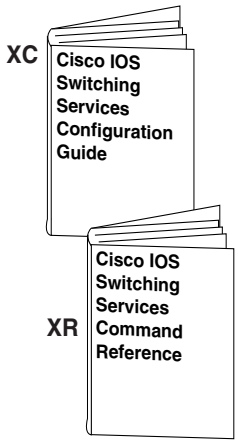
Module BC/B2R:
- DSPU and SNA Service Point
- SNA Switching Services
- Cisco Transaction Connection
- Cisco Mainframe Channel Connection
- CLAW and TCP/IP Offload
- CSNA, CMPC, and CMPC+
- TN3270 Server



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

For Cisco IOS Release 12.2, two previous Release 12.1 guides, *Cisco IOS Dial Services Configuration Guide: Terminal Services* and *Cisco IOS Dial Services Configuration Guide: Network Services*, have been renamed and reorganized into a single book: *Cisco IOS Dial Technologies Configuration Guide*. See Figure 1 for a list of the contents.

For Cisco IOS Release 12.2, the Release 12.1 *Cisco IOS Dial Services Command Reference* has been renamed *Cisco IOS Dial Technologies Command Reference*.

The *Cisco IOS Terminal Services Configuration Guide* and *Cisco IOS Terminal Services Command Reference* were extracted from the 12.1 release of the *Cisco IOS Dial Services Configuration Guide: Terminal Services* and *Cisco IOS Dial Services Command Reference,* and placed in separate books not included in this set.

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.

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

Documentation CD-ROM

Cisco documentation and additional literature are available in a CD-ROM package, which ships with your product. The Documentation CD-ROM is updated monthly and may be more current than printed documentation. The CD-ROM package is available as a single unit or through an annual subscription.

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

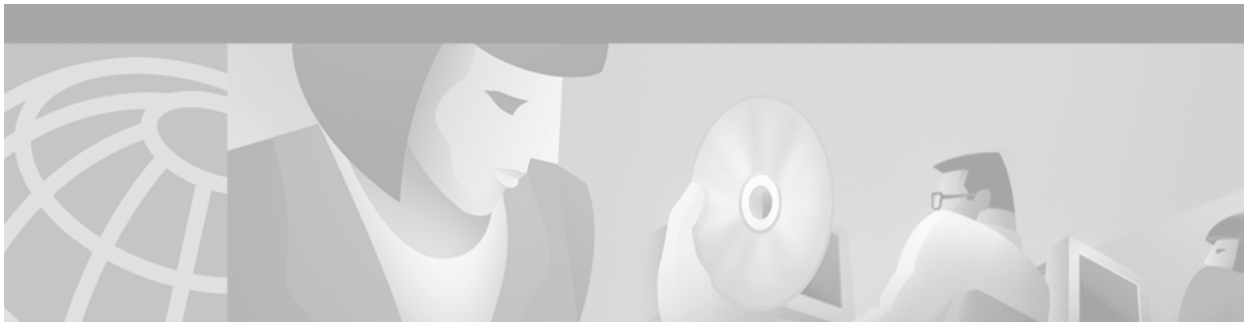
Contacting TAC by Telephone

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:

- Understanding Command Modes
- Getting Help
- Using the no and default Forms of Commands
- Saving Configuration Changes
- Filtering Output from the show and more Commands
- Identifying Supported Platforms

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*, Release 12.2.

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:

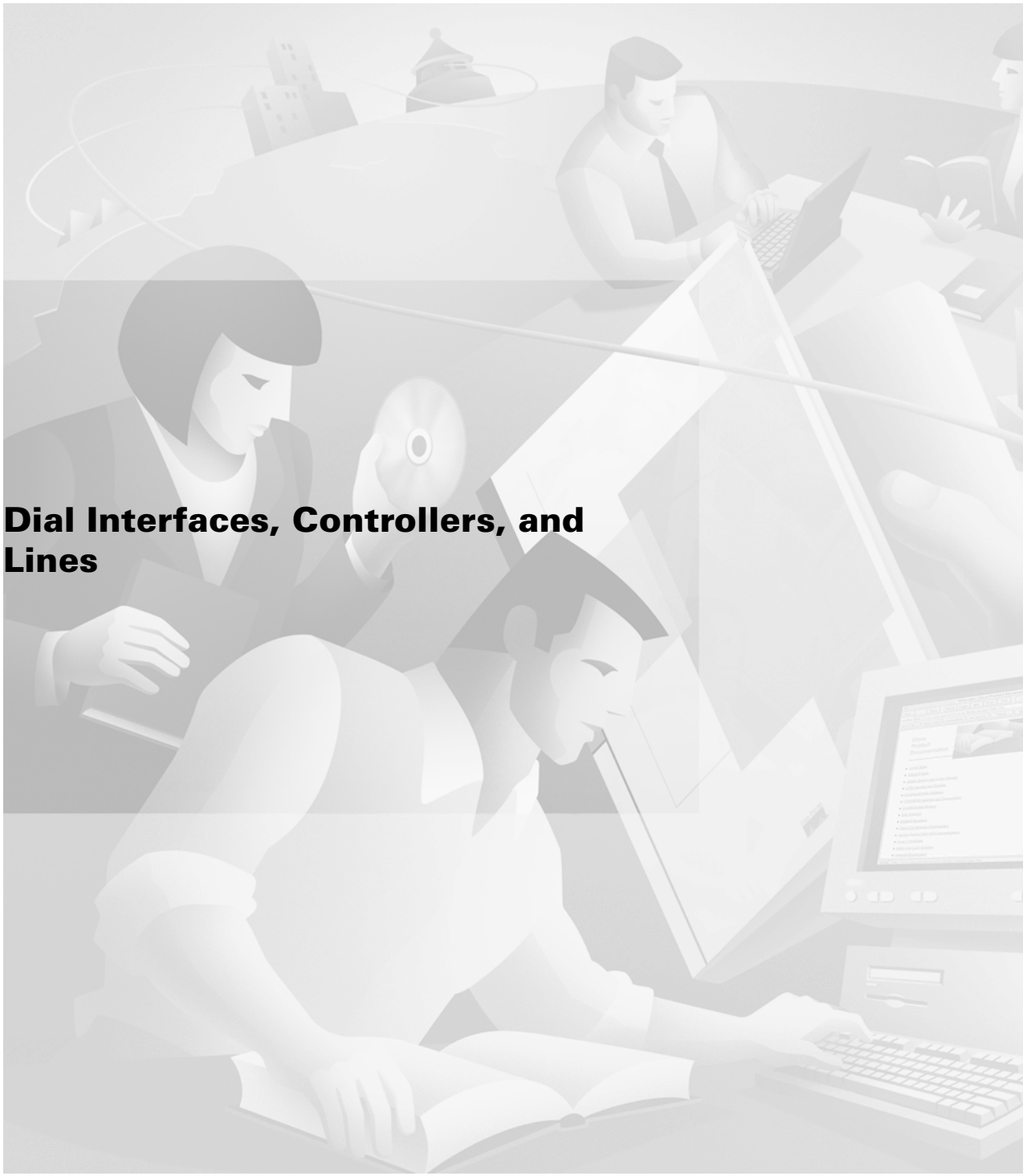
http://www.cisco.com/go/fn

Using Software Release Notes

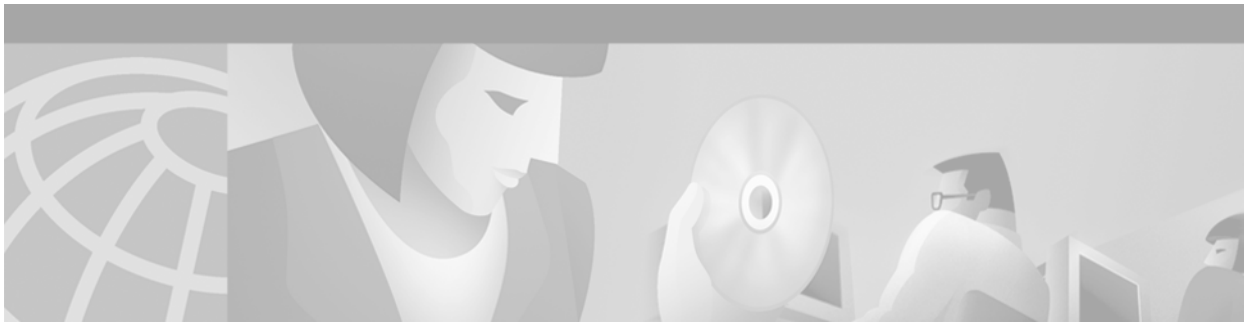
Cisco IOS software releases include release notes that provide the following information:

- 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.



Dial Interfaces, Controllers, and Lines



Overview of Dial Interfaces, Controllers, and Lines

This chapter describes the different types of software constructs, interfaces, controllers, channels, and lines that are used for dial-up remote access. It includes the following main sections:

- Cisco IOS Dial Components
- Logical Constructs
- Logical Interfaces
- Circuit-Switched Digital Calls
- T1 and E1 Controllers
- Non-ISDN Channelized T1 and Channelized E1 Lines
- ISDN Service
- Line Types
- Encapsulation Types

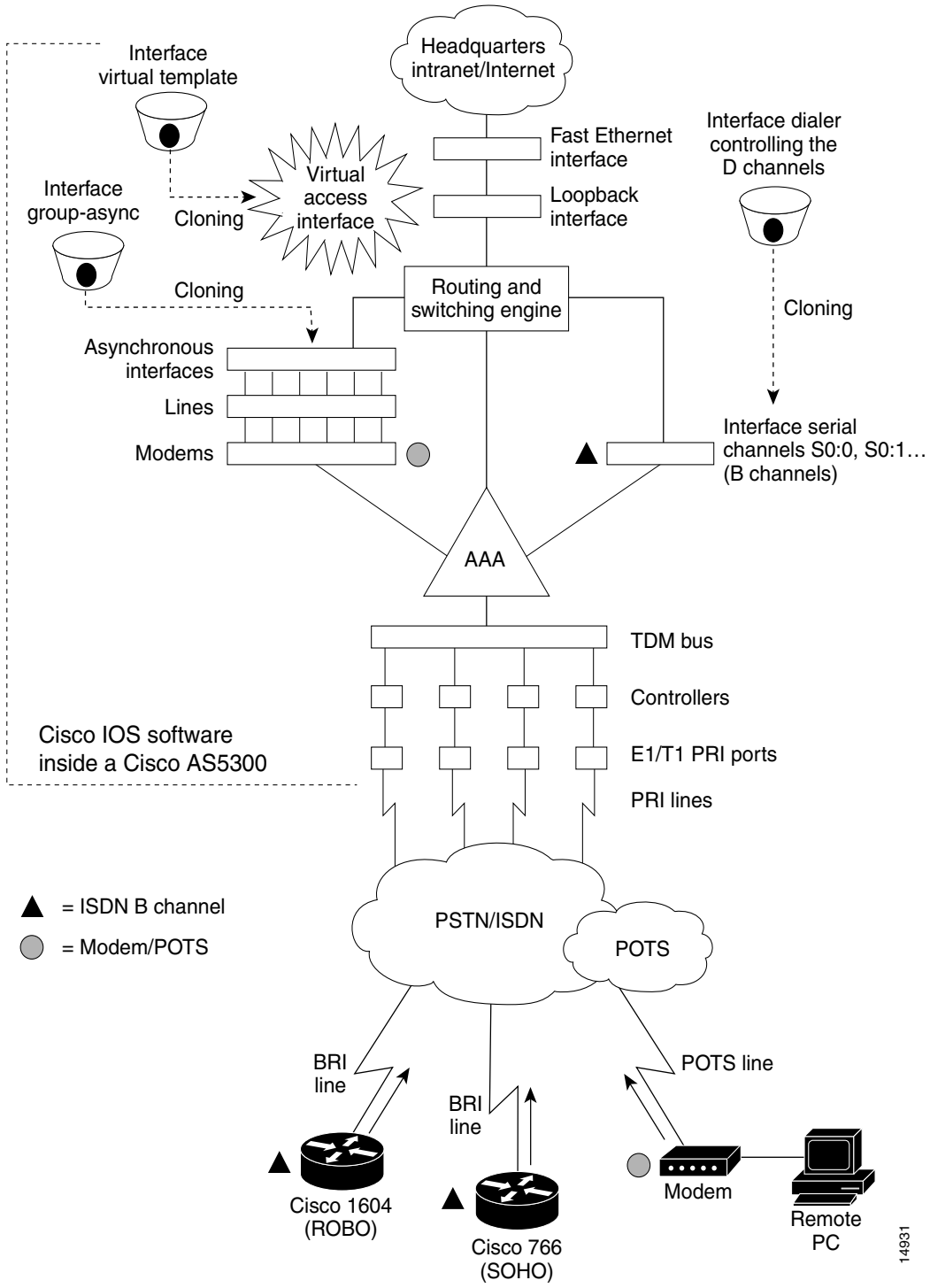
For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Cisco IOS Dial Components

Different components inside Cisco IOS software work together to enable remote clients to dial in and send packets. Figure 2 shows one Cisco AS5300 access server that is receiving calls from a remote office, branch office (ROBO); small office, home office (SOHO); and modem client.

Depending on your network scenario, you may encounter all of the components in Figure 2. For example, you might decide to create a virtual IP subnet by using a loopback interface. This step saves address space. Virtual subnets can exist inside devices that you advertise to your backbone. In turn, IP packets get relayed to remote PCs, which route back to the central site.

Figure 2 Cisco IOS Dial Universe



Logical Constructs

A logical construct stores core protocol characteristics to assign to physical interfaces. No data packets are forwarded to a logical construct. Cisco uses three types of logical constructs in its access servers and routers. These constructs are described in the following sections:

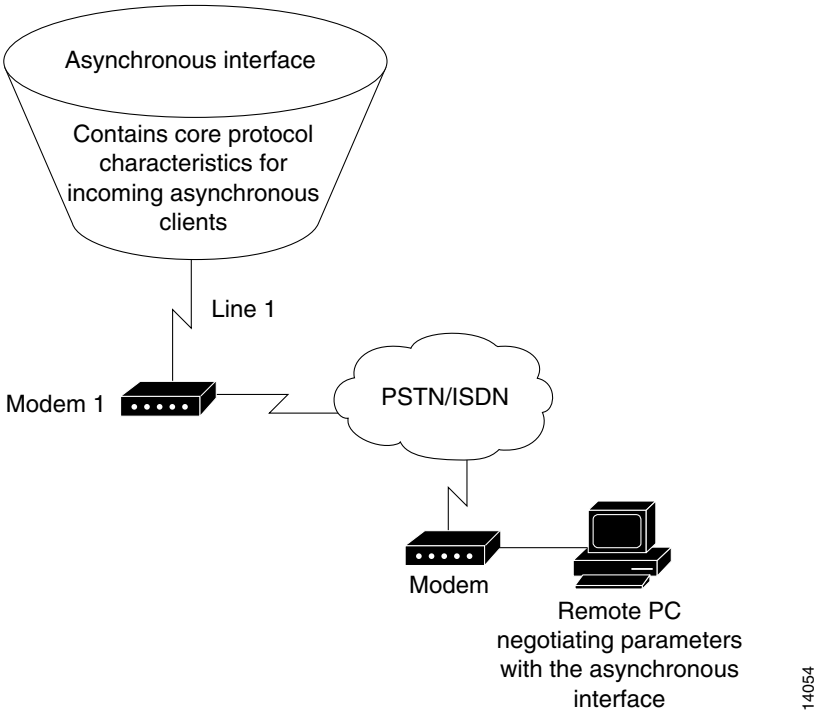
- Asynchronous Interfaces
- Group Asynchronous Interfaces
- Virtual Template Interfaces

Asynchronous Interfaces

An asynchronous interface assigns network protocol characteristics to remote asynchronous clients that are dialing in through physical terminal lines and modems. (See Figure 3.)

Use the **interface async** command to create and configure an asynchronous interface.

Figure 3 Logical Construct for an Asynchronous Interface



To enable clients to dial in, you must configure two asynchronous components: asynchronous lines and asynchronous interfaces. Asynchronous interfaces correspond to physical terminal lines. For example, asynchronous interface 1 corresponds to tty line 1.

Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration configure the physical aspects for the same port.

Specifically, you configure asynchronous interfaces to support PPP connections. An asynchronous interface on an access server or router can be configured to support the following functions:

- Network protocol support such as IP, Internet Protocol Exchange (IPX), or AppleTalk
- Encapsulation support (such as PPP)
- IP client addressing options (default or dynamic)
- IPX network addressing options
- PPP authentication
- ISDN BRI and PRI configuration

For additional information about configuring asynchronous interfaces, see the chapter "Configuring Asynchronous Lines and Interfaces."

Group Asynchronous Interfaces

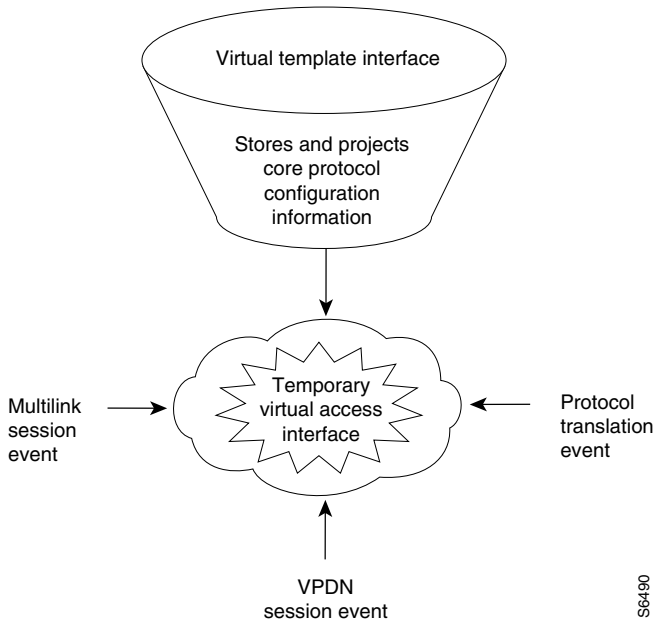
A group asynchronous interface is a parent interface that stores core protocol characteristics and projects them to a specified range of asynchronous interfaces. Asynchronous interfaces clone protocol information from group asynchronous interfaces. No data packets arrive in a group asynchronous interface. By setting up a group asynchronous interface, you also eliminate the need to repeatedly configure identical configuration information across several asynchronous interfaces.

See the "Overview of Modem Interfaces" chapter for more information about group asynchronous interfaces.

Virtual Template Interfaces

A virtual template interface stores protocol configuration information for virtual access interfaces and protocol translation sessions. (See Figure 4.)

Figure 4 Logical Construct for a Virtual Template Interface



Templates for Virtual Access Interfaces

Virtual templates project configuration information to temporary virtual access interfaces triggered by multilink or virtual private dial-up network (VPDN) session events. When a virtual access interface is triggered, the configuration attributes in the virtual template are cloned and the negotiated parameters are applied to the connection.

The following example shows a virtual template interface on a Cisco 7206 router, which is used as a home gateway in a VPDN scenario:

```
Router# configure terminal
Router(config)# interface virtual-template 1
Router(config-if)# ip unnumbered ethernet 2/1
Router(config-if)# peer default ip address pool cisco-pool
Router(config-if)# ppp authentication chap pap
Router(config-if)# exit
Router(config)# vpdn enable
Router(config)# vpdn incoming isp cisco.com virtual-template 1
```

Templates for Protocol Translation

Virtual templates are used to simplify the process of configuring protocol translation to tunnel PPP or Serial Line Internet Protocol (SLIP) across X.25, TCP, and LAT networks. You can create a virtual interface template using the **interface virtual-template** command, and you can use it for one-step and two-step protocol translation. When a user dials in through a vty line and a tunnel connection is established, the router clones the attributes of the virtual interface template onto a *virtual access interface*. This virtual access interface is a temporary interface that supports the protocol configuration specified in the virtual interface template. This virtual access interface is created dynamically and lasts only as long as the tunnel session is active.

The virtual template in the following example explicitly specifies PPP encapsulation. The translation is from X.25 to PPP, which enables tunneling of PPP across an X.25 network.

```
Router# configure terminal
Router(config)# interface virtual-template 1
Router(config-if)# ip unnumbered ethernet 0
Router(config-if)# peer default ip address 172.18.2.131
Router(config-if)# encapsulation ppp
Router(config-if)# exit
Router(config)# translate x25 5555678 virtual-template 1
```

For more information, refer to the chapter “Configuring Protocol Translation and Virtual Asynchronous Devices” in the *Cisco IOS Terminal Services Configuration Guide*.

Logical Interfaces

A logical interface receives and sends data packets and controls physical interfaces. Cisco IOS software provides three logical interfaces used for dial access. These interfaces are described in the following sections:

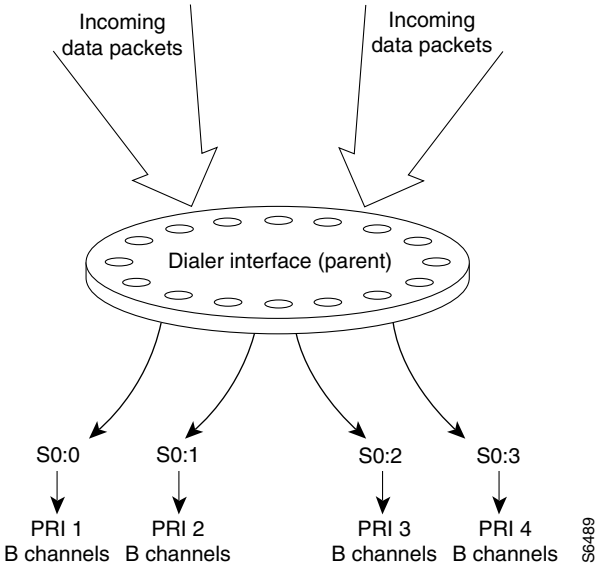
- Dialer Interfaces
- Virtual Access Interfaces
- Virtual Asynchronous Interfaces

Dialer Interfaces

A dialer interface is a parent interface that stores and projects protocol configuration information that is common to all data (D) channels that are members of a dialer rotary group. Data packets pass through dialer interfaces, which in turn initiate dialing for inbound calls. In most cases, D channels get their core protocol intelligence from dialer interfaces.

Figure 5 shows packets coming into a dialer interface, which contains the configuration parameters common to four D channels (shown as S0:0, S0:1, S0:2, and S0:3). All the D channels are members of the same rotary group. Without the dialer interface configuration, each D channel must be manually configured with identical properties. Dialer interfaces condense and streamline the configuration process.

Figure 5 Dialer Interface and Its Neighboring Components



A dialer interface is user configurable and linked to individual B channels, where it delivers data packets to their physical destinations. Dialer interfaces seize physical interfaces to cause packet delivery. If a dialer interface engages in a multilink session, a dialer interface is in control of a virtual access interface, which in turn controls S0:3 or chassis 2 S0:3, for example. A dialer interface is created with the **interface dialer** global configuration command.

The following example shows a fully configured dialer interface:

```
Router# configure terminal
Router(config)# interface dialer 0
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# no ip mroute-cache
Router(config-if)# encapsulation ppp
Router(config-if)# peer default ip address pool dialin_pool
Router(config-if)# dialer in-band
Router(config-if)# dialer-group 1
Router(config-if)# no fair-queue
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap pap callin
Router(config-if)# ppp multilink
```

All the D channels are members of rotary group 1.

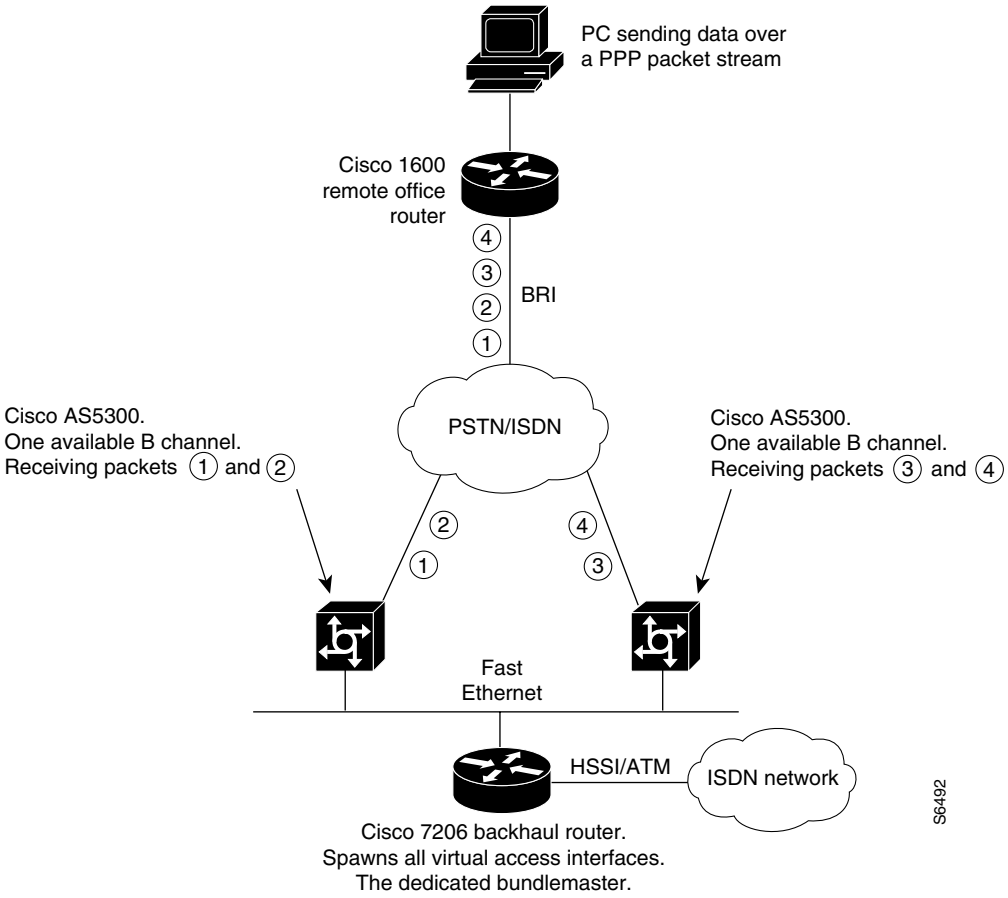
Virtual Access Interfaces

A virtual access interface is a temporary interface that is spawned to terminate incoming PPP streams that have no physical connections. PPP streams, Layer 2 Forwarding Protocol (L2F), and Layer 2 Tunnel Protocol (L2TP) frames that come in on multiple B channels are reassembled on virtual access interfaces. These access interfaces are constructs used to terminate packets.

Virtual access interfaces obtain their set of instructions from virtual interface templates. The attributes configured in virtual templates are projected or cloned to a virtual access interfaces. Virtual access interfaces are not directly user configurable. These interfaces are created dynamically and last only as long as the tunnels or multilink sessions are active. After the sessions end, the virtual access interfaces disappear.

Figure 6 shows how a virtual access interface functions to accommodate a multilink session event. Two physical interfaces on two different access servers are participating in one multilink call from a remote PC. However, each Cisco AS5300 access server has only one B channel available to receive a call. All other channels are busy. Therefore all four packets are equally dispersed across two separate B channels and two access servers. Each Cisco AS5300 access server receives only half the total packets. A virtual access interface is dynamically spawned upstream on a Cisco 7206 backhaul router to receive the multilink protocol, track the multilink frames, and reassemble the packets. The Cisco 7206 router is configured to be the bundle master, which performs all packet assembly and reassembly for both Cisco AS5300 access servers.

Figure 6 Virtual Access Interfaces Used for Multichassis Multilink Session Events

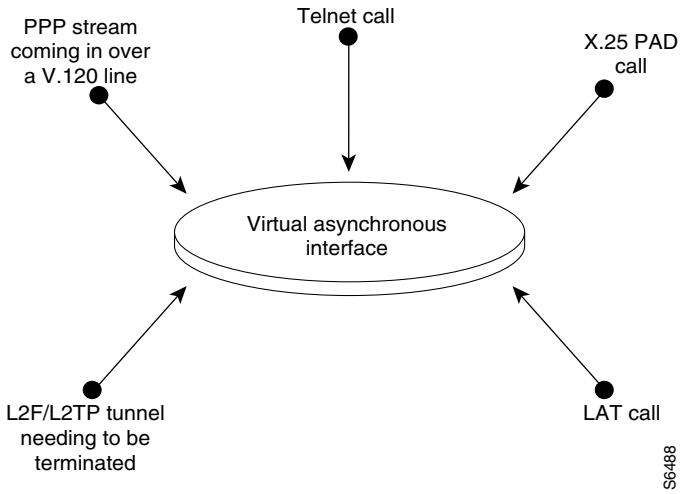


Virtual Asynchronous Interfaces

A virtual asynchronous interface is created on demand to support calls that enter the router through a nonphysical interface. For example, asynchronous character stream calls terminate or land on nonphysical interfaces. These types of calls include inbound Telnet, LAT, PPP over character-oriented protocols (such as V.120 or X.25), and LAPB-TA and PAD calls. A virtual asynchronous interface is also used to terminate L2F/L2TP tunnels, which are often traveling companions with Multilink protocol sessions. Virtual asynchronous interfaces are not user configurable; rather, they are dynamically created and torn down on demand. A virtual asynchronous line is used to access a virtual asynchronous interface.

Figure 7 shows a variety of calls that are terminating on a virtual asynchronous interface. After the calls end, the interface is torn down.

Figure 7 Asynchronous Character Stream Calls Terminating on a Virtual Asynchronous Interface

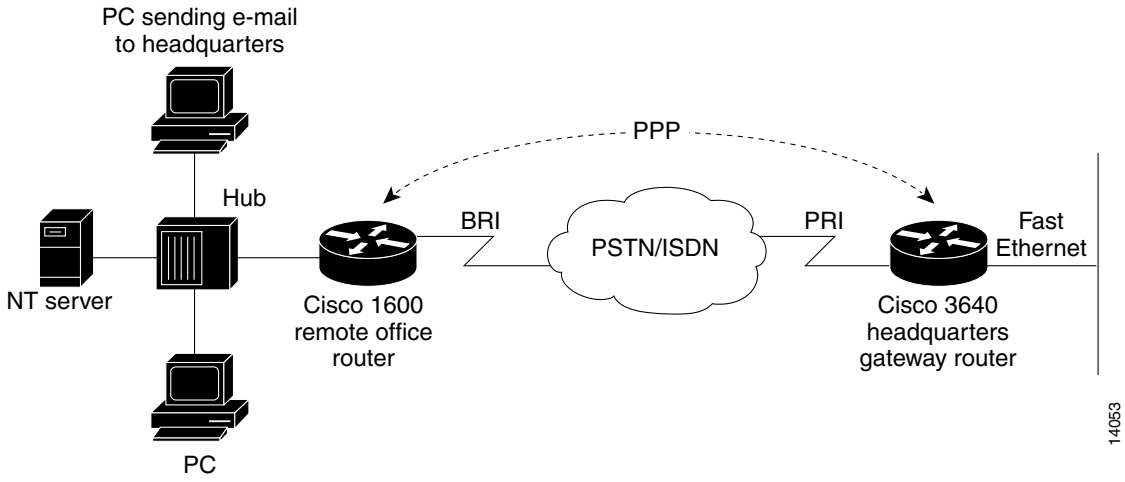


Circuit-Switched Digital Calls

Circuit-switched digital calls are usually ISDN 56-kbps or 64-kbps data calls that use PPP. These calls are initiated by an ISDN router, access server, or terminal adapter that is connected to a client workstation. Individual synchronous serial digital signal level 0 (DS0) bearer (B) channels are used to transport circuit-switched digital calls across WANs. These calls do not transmit across “old world” lines.

Figure 8 shows a Cisco 1600 series remote office router dialing in to a Cisco 3640 router positioned at a headquarters gateway.

Figure 8 Remote Office LAN Dialing In to Headquarters



T1 and E1 Controllers

Cisco controllers negotiate the following parameters between an access server and a central office: line coding, framing, clocking, DS0/time-slot provisioning, and signaling.

Time slots are provisioned to meet the needs of particular network scenarios. T1 controllers have 24 time slots, and E1 controllers have 30 time slots. To support traffic flow for one ISDN PRI line in a T1 configuration, use the **pri-group** command. To support traffic flow for analog calls over a channelized E1 line with recEive and transMit (E&M—also ear and mouth) signaling, use the **cas-group 1 timeslots 1-30 type e&m-fgb** command. Most telephone companies do not support provisioning one trunk for different combinations of time-slot services, though this provisioning is supported on Cisco controllers. On a T1 controller, for example, time slots 1 to 10 could run PRI, time slots 11 to 20 could run channel-associated signaling (CAS), and time slots 21 to 24 could support leased-line grouping.

The following example configures one of four T1 controllers on a Cisco AS5300 access server:

```
Router# configure terminal
Router(config)# controller t1 ?
  <0-3>  Controller unit number
Router(config)# controller t1 0
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# clock source line primary
Router(config-controller)# pri-group timeslots 1-24
Router(config-controller)#
```

This example supports modem calls and circuit-switched digital calls over ISDN PRI.

Non-ISDN Channelized T1 and Channelized E1 Lines

A channelized T1 or channelized E1 line is an analog line that was originally intended to support analog voice calls, but has evolved to support analog data calls. ISDN is not sent across channelized T1 or E1 lines. Channelized T1 and channelized E1 lines are often referred to as CT1 and CE1. These channelized lines are found in “old world,” non-ISDN telephone networks.

The difference between traditional channelized lines (analog) and nonchannelized lines (ISDN) is that channelized lines have no built-in D channel. That is, all 24 channels on a T1 line carry only data. The signaling is in-band or associated to the data channels. Traditional channelized lines do not support digitized data calls (for example, BRI with 2B + D). Channelized lines support a variety of in-band signal types, such as ground start, loop start, wink start, immediate start, E&M, and R2.

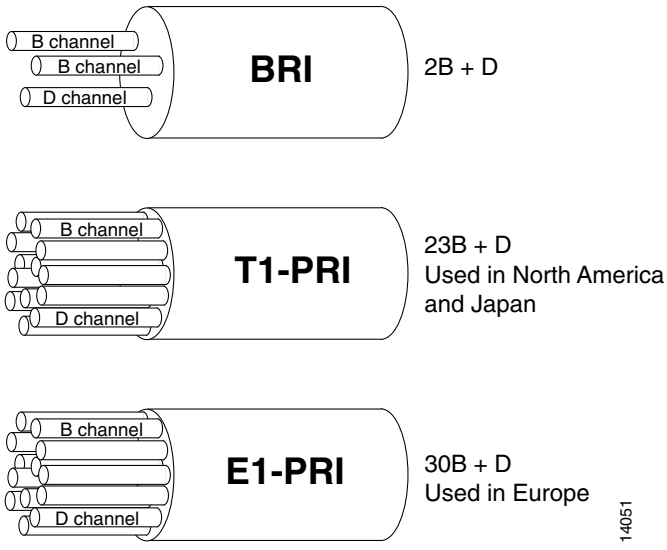
Signaling for channelized lines is configured with the **cas-group** controller configuration command. The following example configures E&M group B signaling on a T1 controller:

```
Router# configure terminal
Router(config)# controller t1 0
Router(config-controller)# cas-group 1 timeslots 1-24 type ?
  e&m-fgb              E & M Type II FGB
  e&m-fgd              E & M Type II FGD
  e&m-immediate-start  E & M Immediate Start
  fxs-ground-start     FXS Ground Start
  fxs-loop-start       FXS Loop Start
  r1-modified          R1 Modified
  sas-ground-start     SAS Ground Start
  sas-loop-start       SAS Loop Start
Router(config-controller)# cas-group 1 timeslots 1-24 type e&m-fgb
Router(config-controller)# framing esf
Router(config-controller)# clock source line primary
```

ISDN Service

Cisco routing devices support ISDN BRI and ISDN PRI. Both media types use B channels and D channels. Figure 9 shows how many B channels and D channels are assigned to each media type.

Figure 9 Logical Relationship of B Channels and D Channels



ISDN BRI

ISDN BRI operates over most of the copper twisted-pair telephone wiring in place. ISDN BRI delivers a total bandwidth of a 144 kbps via three separate channels. Two of the B channels operate at 64 kbps and are used to carry voice, video, or data traffic. The third channel, the D channel, is a 16-kbps signaling channel used to tell the Public Switched Telephone Network (PSTN) how to handle each of the B channels. ISDN BRI is often referred to as "2 B + D."

Enter the **interface bri** command to bring up and configure a single BRI interface, which is the overseer of the 2 B + D channels. The D channel is not user configurable.

The following example configures an ISDN BRI interface on a Cisco 1600 series router. The **isdn spid** command defines the service profile identifier (SPID) number for both B channels. The SPID number is assigned by the ISDN service provider. Not all ISDN lines have SPIDs.

```
Router# configure terminal

Router(config)# interface bri 0
Router(config-if)# isdn spid1 55598760101
Router(config-if)# isdn spid2 55598770101
Router(config-if)# isdn switch-type basic-ni
Router(config-if)# ip unnumbered ethernet 0
Router(config-if)# dialer map ip 172.168.37.40 name hq 5552053
Router(config-if)# dialer load-threshold 70
Router(config-if)# dialer-group 1
Router(config-if)# encapsulation ppp
Router(config-if)# ppp authentication chap pap callin
Router(config-if)# ppp multilink
Router(config-if)# no shutdown
```

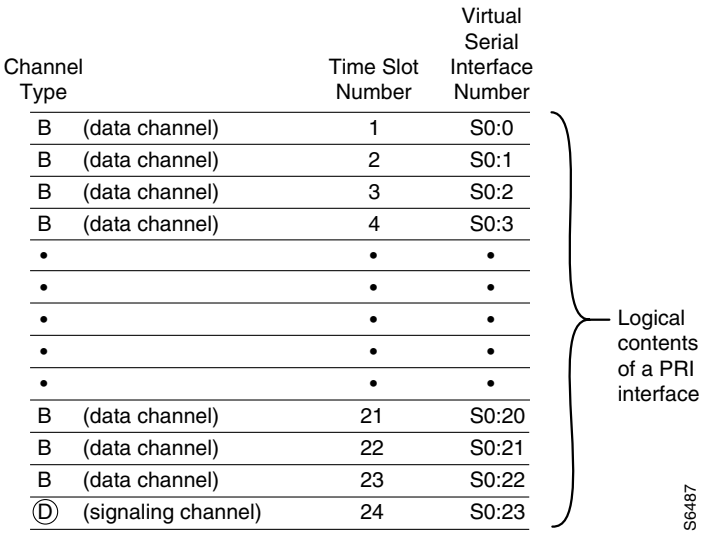
ISDN PRI

ISDN PRI is designed to carry large numbers of incoming ISDN calls at point of presences (POPs) and other large central site locations. All the reliability and performance of ISDN BRI applies to ISDN PRI, but ISDN PRI has 23 B channels running at 64 kbps each and a shared 64 kbps D channel that carries signaling traffic. ISDN PRI is often referred to as "23 B + D" (North America and Japan) or "30 B + D" (rest of the world).

The D channel notifies the central office switch to send the incoming call to particular timeslots 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 directly relayed to the ISDN processor in the router. Enter the **interface serial** command to bring up and configure the D channel, which is user configurable.

Figure 10 shows the logical contents of an ISDN PRI interface used in a T1 network configuration. The logical contents include 23 B channels, 1 D channel, 24 time slots, and 24 virtual serial interfaces (total number of B + D channels).

Figure 10 Logical Relationship of ISDN PRI Components for T1



The following example is for a Cisco AS5300 access server. It configures one T1 controller for ISDN PRI, then configures the neighboring D channel (interface serial 0:23). Controller T1 0 and interface serial 0:23 are both assigned to the first PRI port. The second PRI port is assigned to controller T1 1 and interface serial 1:23, and so on. The second PRI port configuration is not shown in this example. This Cisco AS5300 access server is used as part of a stack group dial-in solution for an Internet service provider.

```
Router# configure terminal

Router(config)# controller t1 0
Router(config-controller)# framing esf
Router(config-controller)# linecode b8zs
Router(config-controller)# clock source line primary
Router(config-controller)# pri-group timeslots 1-24
Router(config-controller)# exit
Router(config)# interface serial 0:23
Router(config-if)# ip unnumbered Loopback 0
Router(config-if)# ip accounting output-packets
Router(config-if)# no ip mroute-cache
Router(config-if)# encapsulation ppp
Router(config-if)# isdn incoming-voice modem
Router(config-if)# dialer-group 1
Router(config-if)# no fair-queue
Router(config-if)# compress stac
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap
Router(config-if)# ppp multilink
Router(config-if)# netbios nbf
```

Line Types

This section describes the different line types used for dial access. It also describes the relationship between lines and interfaces.

Note Cisco devices have four types of lines: console, auxiliary, asynchronous, and virtual terminal. Different routers have different numbers of these line types. Refer to the hardware and software configuration guides that shipped with your device for exact configurations.

Table 3 shows the types of lines that can be configured.

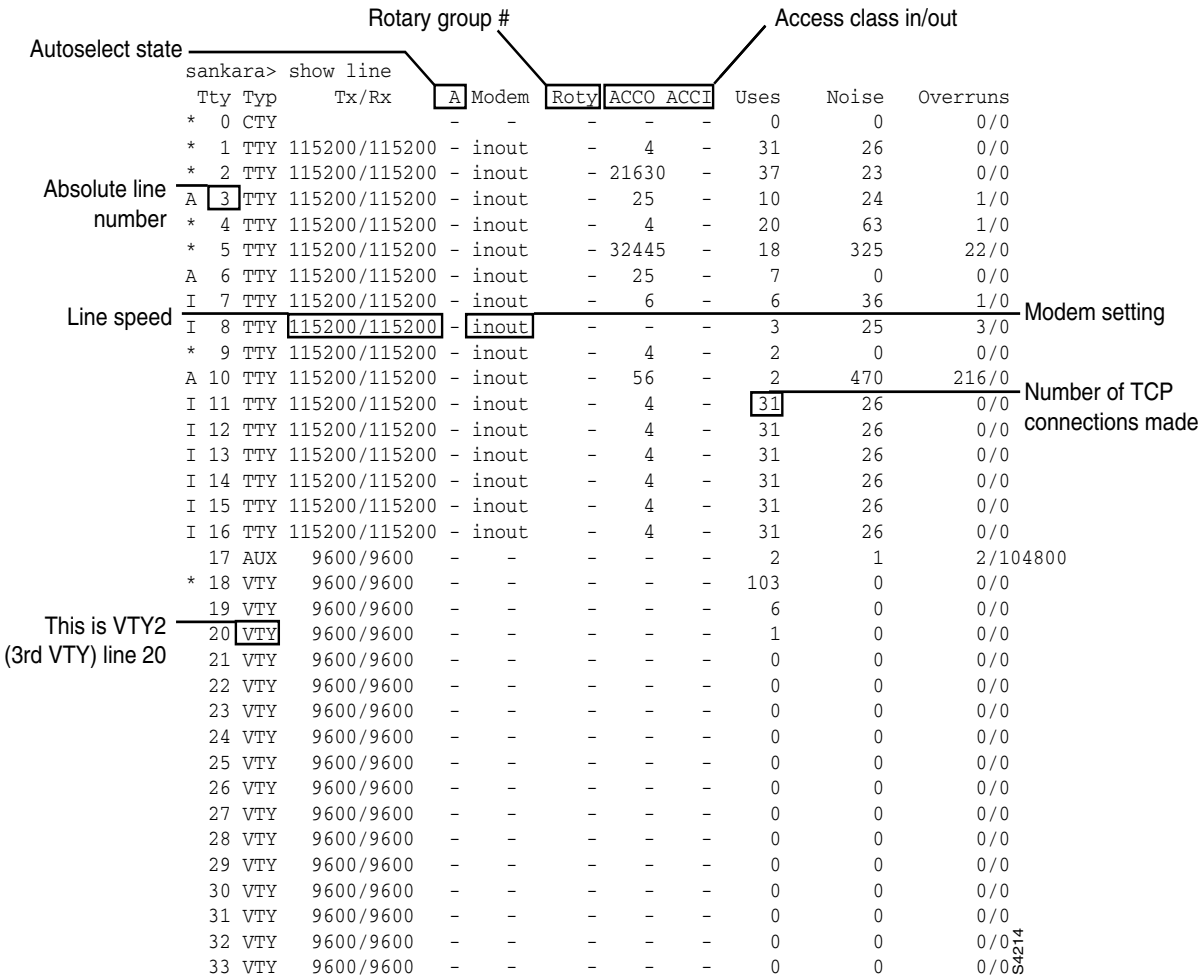
Table 3 Available Line Types

Line Type	Interface	Description	Numbering Rules
CON or CTY	Console	Typically used to log in to the router for configuration purposes.	Line 0.
AUX	Auxiliary	EIA/TIA-232 data terminal equipment (DTE) port used as a backup (tty) asynchronous port. Cannot be used as a second console port.	Last tty line number plus 1.
tty	Asynchronous	Same as asynchronous interface. Used typically for remote-node dial-in sessions that use such protocols as SLIP, PPP, AppleTalk Remote Access (ARA), and XRemote.	The numbering widely varies between platforms. This number is equivalent to the maximum number of modems or asynchronous interfaces supported by your access server or router.[1]
vty	Virtual asynchronous	Used for incoming Telnet, LAT, X.25 PAD, and protocol translation connections into synchronous ports (such as Ethernet and serial interfaces) on the router.	Last tty line number plus 2 through the maximum number of vty lines specified.[2]

1. Enter the **interface line tty ?** command to view the maximum number of tty lines supported.
2. Increase the number of vty lines on a router using the **line vty** global configuration command. Delete vty lines with the **no line vty** *line-number* command. The **line vty** command accepts any line number larger than 5 up to the maximum number of lines supported by your router with its current configuration. Enter the **interface line vty ?** command to view the maximum number of vty lines supported.

Use the **show line** command to see the status of each of the lines available on a router. (See Figure 11.)

Figure 11 Sample Show Line Output Showing CTY, tty, AUX, and vty Line Statistics




```
sankara> show line
 Tty Typ     Tx/Rx      A Modem  Roty ACCO ACCI   Uses    Noise   Overruns
*  0 CTY                -    -      -    -    -      0        0        0/0
*  1 TTY 115200/115200 - inout      -    4    -     31       26        0/0
*  2 TTY 115200/115200 - inout      - 21630   -     37       23        0/0
A  3 TTY 115200/115200 - inout      -   25    -     10       24        1/0
*  4 TTY 115200/115200 - inout      -    4    -     20       63        1/0
*  5 TTY 115200/115200 - inout      - 32445   -     18      325       22/0
A  6 TTY 115200/115200 - inout      -   25    -      7        0        0/0
I  7 TTY 115200/115200 - inout      -    6    -      6       36        1/0
I  8 TTY 115200/115200 - inout      -    -    -      3       25        3/0
*  9 TTY 115200/115200 - inout      -    4    -      2        0        0/0
A 10 TTY 115200/115200 - inout      -   56    -      2      470      216/0
I 11 TTY 115200/115200 - inout      -    4    -     31       26        0/0
I 12 TTY 115200/115200 - inout      -    4    -     31       26        0/0
I 13 TTY 115200/115200 - inout      -    4    -     31       26        0/0
I 14 TTY 115200/115200 - inout      -    4    -     31       26        0/0
I 15 TTY 115200/115200 - inout      -    4    -     31       26        0/0
I 16 TTY 115200/115200 - inout      -    4    -     31       26        0/0
  17 AUX   9600/9600    -    -      -    -    -      2        1        2/104800
* 18 VTY   9600/9600    -    -      -    -    -    103        0        0/0
  19 VTY   9600/9600    -    -      -    -    -      6        0        0/0
  20 VTY   9600/9600    -    -      -    -    -      1        0        0/0
  21 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  22 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  23 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  24 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  25 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  26 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  27 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  28 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  29 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  30 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  31 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  32 VTY   9600/9600    -    -      -    -    -      0        0        0/0
  33 VTY   9600/9600    -    -      -    -    -      0        0        0/0
```

Relationship Between Lines and Interfaces

The following sections describe the relationship between lines and interfaces:

- Asynchronous Interfaces and Physical Terminal Lines
- Synchronous Interfaces and Virtual Terminal Lines

Asynchronous Interfaces and Physical Terminal Lines

Asynchronous interfaces correspond to physical terminal lines. Commands entered in asynchronous interface mode let you configure protocol-specific parameters for asynchronous interfaces; commands entered in line configuration mode let you configure the physical aspects of the line port.

For example, to enable IP resources to dial in to a network through a Cisco 2500 series access server, configure the lines and asynchronous interfaces as follows.

- Configure the physical aspect of a line that leads to a port. You might enter the following commands to configure lines 1 through 16 (asynchronous physical terminal lines on a Cisco 2511 access server):

```
line 1 16
 login local
 modem inout
 speed 115200
 flowcontrol hardware
 ! Configures the line to autosense PPP; physical line attribute.
 autoselect ppp
```

- On asynchronous interface 1, you configure your protocol-specific commands. You might enter the following commands:

```
interface async 1
 encapsulation ppp
 async mode interactive
 async dynamic address
 async dynamic routing
 async default ip address 192.168.16.132
 ppp authentication chap
```

The remote node services SLIP, PPP, and XRemote are configured in asynchronous interface mode. ARA is configured in line configuration mode on virtual terminal lines or physical terminal lines.

Synchronous Interfaces and Virtual Terminal Lines

Virtual terminal lines provide access to the router through a synchronous interface. Virtual terminal lines do not correspond to synchronous interfaces in the same way that physical terminal lines correspond to asynchronous interfaces because vty lines are created dynamically on the router, whereas physical terminal lines are static physical ports. When a user connects to the router on a vty line, that user is connecting into a *virtual* port on an interface. You can have multiple virtual ports for each synchronous interface.

For example, several Telnet connections can be made to an interface (such as an Ethernet or serial interface).

The number of virtual terminal lines available on a router is defined using the **line vty** *number-of-lines* global configuration command.

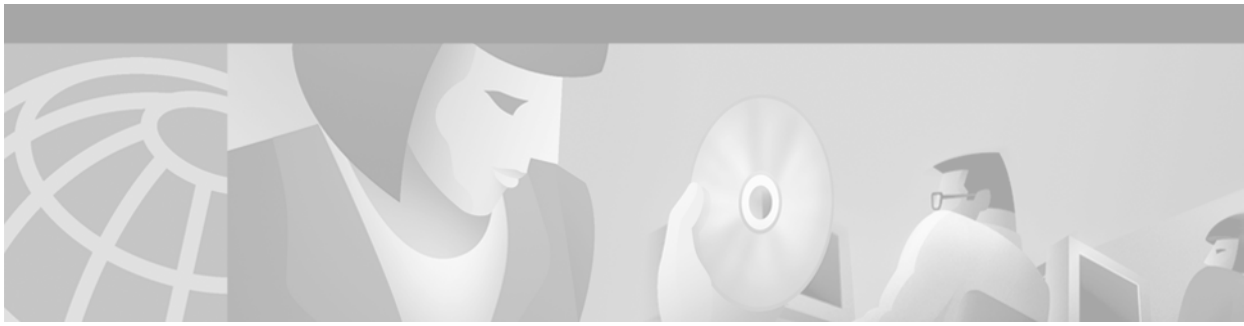
Encapsulation Types

Synchronous serial interfaces default to High-Level Data Link Control (HDLC) encapsulation, and asynchronous serial interfaces default to SLIP encapsulation. Cisco IOS software provides a long list of encapsulation methods that can be set on the interface to change the default encapsulation method. See the *Cisco IOS Interface Command Reference* for a complete list and description of these encapsulation methods.

The following list summarizes the encapsulation commands available for serial interfaces used in dial configurations:

- **encapsulation frame-relay**—Frame Relay
- **encapsulation hdlc**—HDLC protocol
- **encapsulation lapb**—X.25 LAPB DTE operation
- **encapsulation ppp**—PPP
- **encapsulation slip**—SLIP

To use SLIP or PPP encapsulation, the router or access server must be configured with an IP routing protocol or with the **ip host-routing** command.



Configuring Asynchronous Lines and Interfaces

This chapter describes how to configure asynchronous line features in the following main sections:

- How to Configure Asynchronous Interfaces and Lines
- How to Configure Other Asynchronous Line and Interface Features
- Configuration Examples for Asynchronous Interfaces and Lines

Perform these tasks, as required, for your particular network.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

How to Configure Asynchronous Interfaces and Lines

To configure an asynchronous interface, perform the tasks described in the following sections as required:

- Configuring a Typical Asynchronous Interface (As required)
- Creating a Group Asynchronous Interface (As required)
- Configuring Asynchronous Rotary Line Queueing (As required)
- Configuring Autoselect (As required)

Configuring a Typical Asynchronous Interface

To configure an asynchronous interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface async number`	Brings up a single asynchronous interface and enters interface configuration mode.
Step 2	`Router(config-if)# description description`	Provides a description for the interface.
Step 3	`Router(config-if)# ip address address mask`	Specifies an IP address.
Step 4	`Router(config-if)# encapsulation ppp`	Enables PPP to run on the asynchronous interfaces in the group.
Step 5	`Router(config-if)# async default routing`	Enables the router to pass routing updates to other routers over the AUX port configured as an asynchronous interface.
Step 6	`Router(config-if)# async mode dedicated`	Places a line into dedicated asynchronous mode using Serial Line Internet Protocol (SLIP) or PPP encapsulation.
Step 7	`Router(config-if)# dialer in-band`	Specifies that dial-on-demand routing (DDR) is to be supported.
Step 8	`Router(config-if)# dialer map protocol next-hop-address`	Configures a serial interface to call one or multiple sites or to receive calls from multiple sites.
Step 9	`Router(config-if)# dialer-group`	Controls access by configuring an interface to belong to a specific dialing group.
Step 10	`Router(config-if)# ppp authentication chap pap list-name`	Enables Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) authentication on the interface. Replace the *list-name* variable with a specified authentication list name.[1]
Step 11	`Router(config-if)# exit`	Return to global configuration mode.

1. To create a string used to name the following list of authentication methods tried when a user logs in, refer to the **aaa authentication ppp** command. Authentication methods include RADIUS, TACACS+, and Kerberos.

The "Interface and Line Configuration Examples" and "Asynchronous Interface As the Only Network Interface Example" sections later in this chapter contain examples of how to configure an asynchronous interface.

Monitoring and Maintaining Asynchronous Connections

This section describes the following monitoring and maintenance tasks that you can perform on asynchronous interfaces:

- Monitoring and maintaining asynchronous activity
- Debugging asynchronous interfaces
- Debugging PPP

To monitor and maintain asynchronous activity, use the following commands in privileged EXEC mode as needed:

Command	Purpose
`Router# clear line line-number`	Returns a line to its idle state.
`Router# show async bootp`	Displays parameters that have been set for extended BOOTP requests.
`Router# show async status`	Displays statistics for asynchronous interface activity.
`Router# show line [line-number]`	Displays the status of asynchronous line connections.

To debug asynchronous interfaces, use the following debug command in privileged EXEC mode:

Command	Purpose
`Router# debug async {framing | state | packets}`	Displays errors, changes in interface state, and log input and output.

To debug PPP links, use the following debug commands in privileged EXEC mode as needed:

Command	Purpose
`Router# debug ppp negotiation`	Enables debugging of PPP protocol negotiation process.
`Router# debug ppp error`	Displays PPP protocol errors.
`Router# debug ppp packet`	Displays PPP packets sent and received.
`Router# debug ppp chap`	Displays errors encountered during remote or local system authentication.

Creating a Group Asynchronous Interface

Create a group asynchronous interface to project a set of core protocol characteristics to a range of asynchronous interfaces. Configuring the asynchronous interfaces as a group saves you time. Analog modem calls cannot enter the access server without this configuration.

To configure a group asynchronous interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface async number`	Brings up a single asynchronous interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered loopback number`	Configures the asynchronous interfaces as unnumbered and assigns the IP address of the loopback interface to them to conserve IP addresses.[1]
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP to run on the asynchronous interfaces in the group.

	Command	Purpose
Step 4	`Router(config-if)# async mode interactive`	Configures interactive mode on the asynchronous interface.
Step 5	`Router(config-if)# ppp authentication chap pap` *list-name*	Enables CHAP and PAP authentication on the interface. Replace the *list-name* variable with a specified authentication list name.[2]
Step 6	`Router(config-if)# peer default ip address pool` *poolname*	Assigns dial-in clients IP addresses from an address pool.[3]
Step 7	`Router(config-if)# no cdp enable`	Disables the Cisco Discovery Protocol (CDP) on the interface.
Step 8	`Router(config-if)# group-range` *low-end-of-range high-end-of-range*	Specifies the range of asynchronous interfaces to include in the group, which is usually equal to the number of modems you have in the access server.
Step 9	`Router(config-if)# exit`	Returns to global configuration mode.

1. You can also specify the Ethernet interface to conserver address space. In this case, enter the **ip unnumbered ethernet 0** command.
2. To create a string used to name the following list of authentication methods tried when a user logs in, refer to the **aaa authentication ppp** command. Authentication methods include RADIUS, TACACS+, and Kerberos.
3. To create an IP address pool, refer to the **ip local pool** global configuration command.

The "Group and Member Asynchronous Interface Examples" section later in this chapter contains an example of how to configure a group interface.

Verifying the Group Interface Configuration

To verify the group interface configuration and check if one of the asynchronous interfaces is up, use the **show interface async** command:

```
Router# show interface async 1

Async1 is up, line protocol is up
modem(slot/port)=1/0, csm_state(0x00000204)=CSM_IC4_CONNECTED, bchan_num=18
modem_status(0x0002): VDEV_STATUS_ACTIVE_CALL.

  Hardware is Async Serial
  Interface is unnumbered.  Using address of FastEthernet0 (10.1.1.10)
  MTU 1500 bytes, BW 115 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/5, 0 drops; input queue 1/5, 0 drops
  5 minute input rate 37000 bits/sec, 87 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     31063 packets input, 1459806 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
     33 packets output, 1998 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

If you are having trouble, enter one of the following **debug** commands and then send a call into the access server. Interpret the output and make configuration changes accordingly.

- **undebug all**
- **debug ppp negotiation**
- **debug ppp authentication**
- **debug modem**
- **debug ip peer**

```
Router# undebug all
All possible debugging has been turned off
Router# debug ppp negotiation
PPP protocol negotiation debugging is on
Router# debug ppp authentication
PPP authentication debugging is on
Router# debug modem
Modem control/process activation debugging is on
Router# debug ip peer
IP peer address activity debugging is on
Router# show debug
General OS:
  Modem control/process activation debugging is on
Generic IP:
  IP peer address activity debugging is on
PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
Router#
*Mar  1 21:34:56.958: tty4: DSR came up
*Mar  1 21:34:56.962: tty4: Modem: IDLE->READY
*Mar  1 21:34:56.970: tty4: EXEC creation
*Mar  1 21:34:56.978: tty4: set timer type 10, 30 seconds
*Mar  1 21:34:59.722: tty4: Autoselect(2) sample 7E
*Mar  1 21:34:59.726: tty4: Autoselect(2) sample 7EFF
*Mar  1 21:34:59.730: tty4: Autoselect(2) sample 7EFF7D
*Mar  1 21:34:59.730: tty4: Autoselect(2) sample 7EFF7D23
*Mar  1 21:34:59.734: tty4 Autoselect cmd: ppp negotiate
*Mar  1 21:34:59.746: tty4: EXEC creation
*Mar  1 21:34:59.746: tty4: create timer type 1, 600 seconds
*Mar  1 21:34:59.786: ip_get_pool: As4: using pool default
*Mar  1 21:34:59.790: ip_get_pool: As4: returning address = 172.20.1.101
*Mar  1 21:34:59.794: tty4: destroy timer type 1 (OK)
*Mar  1 21:34:59.794: tty4: destroy timer type 0
*Mar  1 21:35:01.798: %LINK-3-UPDOWN: Interface Async4, changed state to up
*Mar  1 21:35:01.834: As4 PPP: Treating connection as a dedicated line
*Mar  1 21:35:01.838: As4 PPP: Phase is ESTABLISHING, Active Open
*Mar  1 21:35:01.842: As4 LCP: O CONFREQ [Closed] id 1 len 25
*Mar  1 21:35:01.846: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:01.850: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:01.854: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:01.854: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:01.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.718: As4 LCP: I CONFREQ [REQsent] id 3 len 23
*Mar  1 21:35:02.722: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.726: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.726: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.730: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.730: As4 LCP:    Callback 6  (0x0D0306)
*Mar  1 21:35:02.738: As4 LCP: O CONFREJ [REQsent] id 3 len 7
*Mar  1 21:35:02.738: As4 LCP:    Callback 6  (0x0D0306)
*Mar  1 21:35:02.850: As4 LCP: I CONFREQ [REQsent] id 4 len 20
```

```
*Mar  1 21:35:02.854: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.854: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.858: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:02.862: As4 LCP: O CONFACK [REQsent] id 4 len 20
*Mar  1 21:35:02.866: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:02.870: As4 LCP:    MagicNumber 0x00472467 (0x050600472467)
*Mar  1 21:35:02.870: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:02.874: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.842: As4 LCP: TIMEout: State ACKsent
*Mar  1 21:35:03.842: As4 LCP: O CONFREQ [ACKsent] id 2 len 25
*Mar  1 21:35:03.846: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:03.850: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:03.854: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:03.854: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:03.858: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.962: As4 LCP: I CONFACK [ACKsent] id 2 len 25
*Mar  1 21:35:03.966: As4 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Mar  1 21:35:03.966: As4 LCP:    AuthProto CHAP (0x0305C22305)
*Mar  1 21:35:03.970: As4 LCP:    MagicNumber 0x64E923A8 (0x050664E923A8)
*Mar  1 21:35:03.974: As4 LCP:    PFC (0x0702)
*Mar  1 21:35:03.974: As4 LCP:    ACFC (0x0802)
*Mar  1 21:35:03.978: As4 LCP: State is Open
*Mar  1 21:35:03.978: As4 PPP: Phase is AUTHENTICATING, by this end
*Mar  1 21:35:03.982: As4 CHAP: O CHALLENGE id 1 len 26 from "nas-1"
*Mar  1 21:35:04.162: As4 CHAP: I RESPONSE id 1 len 26 from "krist"
*Mar  1 21:35:04.170: As4 AUTH: Started process 0 pid 47
*Mar  1 21:35:04.182: As4 CHAP: O SUCCESS id 1 len 4
*Mar  1 21:35:04.186: As4 PPP: Phase is UP
*Mar  1 21:35:04.190: As4 IPCP: O CONFREQ [Not negotiated] id 1 len 10
*Mar  1 21:35:04.194: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:04.202: As4 CDPCP: O CONFREQ [Closed] id 1 len 4
*Mar  1 21:35:04.282: As4 IPCP: I CONFREQ [REQsent] id 1 len 40
*Mar  1 21:35:04.282: As4 IPCP:    CompressType VJ 15 slots CompressSlotID (0x02
06002D0F01)
*Mar  1 21:35:04.286: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:04.290: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:04.294: As4 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
*Mar  1 21:35:04.298: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:04.302: As4 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
*Mar  1 21:35:04.306: As4 IPCP: O CONFREJ [REQsent] id 1 len 10
*Mar  1 21:35:04.310: As4 IPCP:    CompressType VJ 15 slots CompressSlotID (0x02
06002D0F01)
*Mar  1 21:35:04.314: As4 CCP: I CONFREQ [Not negotiated] id 1 len 15
*Mar  1 21:35:04.318: As4 CCP:    MS-PPC supported bits 0x00000001 (0x1206000000
01)
*Mar  1 21:35:04.318: As4 CCP:    Stacker history 1 check mode EXTENDED (0x11050
00104)
*Mar  1 21:35:04.322: As4 LCP: O PROTREJ [Open] id 3 len 21 protocol CCP
*Mar  1 21:35:04.326: As4 LCP:  (0x80FD0101000F12060000000111050001)
*Mar  1 21:35:04.330: As4 LCP:  (0x04)
*Mar  1 21:35:04.334: As4 IPCP: I CONFACK [REQsent] id 1 len 10
*Mar  1 21:35:04.338: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:04.342: As4 LCP: I PROTREJ [Open] id 5 len 10 protocol CDPCP (0x82
0701010004)
*Mar  1 21:35:04.342: As4 CDPCP: State is Closed
*Mar  1 21:35:05.186: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async4, ch
anged state to up
*Mar  1 21:35:05.190: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:35:05.190: As4 PPP: Trying to negotiate NCP for Link cdp
*Mar  1 21:35:05.194: As4 CDPCP: State is Closed
*Mar  1 21:35:05.198: As4 CDPCP: TIMEout: State Closed
*Mar  1 21:35:05.202: As4 CDPCP: State is Listen
*Mar  1 21:35:06.202: As4 IPCP: TIMEout: State ACKrcvd
```

```
*Mar  1 21:35:06.206: As4 IPCP: O CONFREQ [ACKrcvd] id 2 len 10
*Mar  1 21:35:06.206: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:06.314: As4 IPCP: I CONFACK [REQsent] id 2 len 10
*Mar  1 21:35:06.318: As4 IPCP:    Address 172.20.1.2 (0x0306AC140102)
*Mar  1 21:35:07.274: As4 IPCP: I CONFREQ [ACKrcvd] id 2 len 34
*Mar  1 21:35:07.278: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:07.282: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:07.286: As4 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
*Mar  1 21:35:07.286: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:07.290: As4 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
*Mar  1 21:35:07.294: As4 IPCP: O CONFNAK [ACKrcvd] id 2 len 34
*Mar  1 21:35:07.298: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.302: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.306: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.310: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.314: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.426: As4 IPCP: I CONFREQ [ACKrcvd] id 3 len 34
*Mar  1 21:35:07.430: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.434: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.438: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.442: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.446: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.446: ip_get_pool: As4: validate address = 172.20.1.101
*Mar  1 21:35:07.450: ip_get_pool: As4: using pool default
*Mar  1 21:35:07.450: ip_get_pool: As4: returning address = 172.20.1.101
*Mar  1 21:35:07.454: set_ip_peer_addr: As4: address = 172.20.1.101 (3) is redun
dant
*Mar  1 21:35:07.458: As4 IPCP: O CONFACK [ACKrcvd] id 3 len 34
*Mar  1 21:35:07.462: As4 IPCP:    Address 172.20.1.101 (0x0306AC140165)
*Mar  1 21:35:07.466: As4 IPCP:    PrimaryDNS 172.20.5.100 (0x8106AC140564)
*Mar  1 21:35:07.470: As4 IPCP:    PrimaryWINS 172.20.5.101 (0x8206AC140565)
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryDNS 172.20.6.100 (0x8306AC140664)
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryWINS 172.20.6.101 (0x8406AC140665)
*Mar  1 21:35:07.478: As4 IPCP: State is Open
*Mar  1 21:35:07.490: As4 IPCP: Install route to 172.20.1.101
*Mar  1 21:35:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:36:12.614: tty0: timer type 1 expired
*Mar  1 21:36:12.614: tty0:  Exec timer (continued)
*Mar  1 21:36:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:37:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:38:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:39:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:40:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:41:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:42:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
*Mar  1 21:43:25.038: As4 PPP: Unsupported or un-negotiated protocol. Link cdp
```

Configuring Asynchronous Rotary Line Queueing

The Cisco IOS Asynchronous Rotary Line Queueing feature allows Telnet connection requests to busy asynchronous rotary groups to be queued so that users automatically obtain the next available line, rather than needing to try repeatedly to open a Telnet connection. The Cisco IOS software sends a periodic message to the user to update progress in the connection queue.

This feature allows users to make effective use of the asynchronous rotary groups on a Cisco router to access legacy mainframes or other serial devices with a limited number of asynchronous ports that might be used by a large number of users. Users that are unable to make a Telnet connection on the first attempt are assured of eventual success in an orderly process. They are no longer required to guess when a line might be available and to retry manually again and again.

Connections are authenticated using the method specified for the line configurations for the asynchronous rotary group. If a connection is queued, authentication is done prior to queueing and no authentication is done when the connection is later established.

Make sure you comply with the following requirements when configuring asynchronous rotary line queueing:

- Configure more virtual terminal lines than will ever be used by waiting asynchronous rotary connection attempts. Even when the queue is at its maximum, there must be at least one virtual terminal line available so that system operators or network administrators can use Telnet to access the router to show, debug, or configure system performance.
- When adding lines to a rotary group, all lines must be either queued or not queued. A mixture of queued and unenqueued lines in the same rotary group is not supported and can result in unexpected behavior.
- All lines within a queued rotary group need to use the same authentication method. Using different authentication methods within the same rotary group can result in unexpected behavior.

To configure asynchronous rotary line queueing, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router (config)# line [aux | console | tty | vty] line-number [ending-line-number]`	Starts line configuration mode on the line type and numbers specified.
Step 2	`Router(config-line)# rotary group [queued | round-robin]`	Enables asynchronous rotary line queueing on the designated line or group of lines. The optional **round-robin** keyword selects a round-robin port selection algorithm instead of the default (**queued**) linear port selection algorithm.

See the "Rotary Group Examples" section for configuration examples.

Verifying Asynchronous Rotary Line Queueing

To verify operation of asynchronous rotary line queueing, perform the following tasks:

- Use the **show line** command in EXEC mode to check the status of the vty lines.
- Use the **show line async-queue** command in EXEC mode to check the status of queued connection requests.

Troubleshooting Asynchronous Rotary Lines

If asynchronous rotary line queueing is not operating correctly, use the following **debug** commands in privileged EXEC mode to determine where the problem may lie:

- **debug async async-queue**
- **debug ip tcp transactions**
- **debug modem**

Refer to the *Cisco IOS Debug Command Reference* for information about these commands.

Monitoring and Maintaining Asynchronous Rotary Line Queues

To display queued lines and to remove lines from the queue, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# show line async-queue rotary-group`	Displays which lines are queued.
`Router# clear line async-queue rotary-group`	Clears all rotary queues or the specified rotary queue. If the *rotary-group* argument is not specified, all rotary queues are removed.

Configuring Autoselect

Autoselect is used by the access server to sense the protocol being received on an incoming line and to launch the appropriate protocol. Autoselect can be used for AppleTalk Remote Access (ARA), PPP, or SLIP.

When using Autoselect, "login" authentication is bypassed, so if security is required, it must be performed at the protocol level, that is, the AppleTalk Remote Access Protocol (ARAP) or PPP authentication. SLIP does not offer protocol layer authentication.

To configure the Cisco IOS software to allow an ARA, PPP, or SLIP session to start automatically, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autoselect {arap | ppp | slip | during login}`	Configures a line to automatically start an ARA, PPP, or SLIP session.

The **autoselect** command enables the Cisco IOS software to start a process automatically when a start character is received.

The **autoselect** command bypasses the login prompt and enables the specified session to begin automatically. However, when the **autoselect** command is entered with the **during login** keyword, the username or password prompt appears without the need to press the Return key; thus "login" users will get a prompt right away without needing to press the Return key. While the username or password prompt is displayed, you can choose either to answer these prompts or to send packets from an autoselected protocol.

Normally a router avoids line and modem noise by clearing the initial data received within the first one or two seconds. However, when the autoselect PPP feature is configured, the router flushes characters initially received and then waits for more traffic. This flush causes timeout problems with applications that send only one carriage return. To ensure that the input data sent by a modem or other asynchronous device is not lost after line activation, enter the **flush-at-activation** line configuration command.

Note When the **autoselect** command is used, the activation character should be set to the default Return, and exec-character-bits should be set to 7. If you change these defaults, the application cannot recognize the activation request.

See the "High-Density Dial-In Solution Using Autoselect and EXEC Control Example" section for an example that makes use of the autoselect feature.

Verifying Autoselect PPP

The following trace appears when the **debug modem** and **debug ppp negotiation** commands are enabled. As PPP calls pass through the access server, you should see this output.

When autoselect is used, "login" authentication is bypassed. If security is required, it must be performed at the protocol level (that is, ARAP or PPP authentication). SLIP does not offer protocol layer authentication.

```
22:21:02: TTY1: DSR came up
22:21:02: tty1: Modem: IDLE->READY
22:21:02: TTY1: Autoselect started
22:21:05: TTY1: Autoselect sample 7E
22:21:05: TTY1: Autoselect sample 7EFF
22:21:05: TTY1: Autoselect sample 7EFF7D
22:21:05: TTY1 Autoselect cmd: ppp default
22:21:05: TTY1: EXEC creation
%LINK-3-UPDOWN: Interface Async1, changed state to up
22:21:07: ppp: sending CONFREQ, type = 2 (CI_ASYNCMAP), value = A0000
22:21:07: ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = 23BE13AA
22:21:08: PPP Async1: state = REQSENT fsm_rconfack(0xC021): rcvd id 0x11
22:21:08: ppp: config ACK received, type = 2 (CI_ASYNCMAP), value = A0000
22:21:08: ppp: config ACK received, type = 5 (CI_MAGICNUMBER), value = 23BE13AA
22:21:08: ppp: config ACK received, type = 7 (CI_PCOMPRESSION)
22:21:08: ppp: config ACK received, type = 8 (CI_ACCOMPRESSION)
22:21:08: PPP Async1: received config for type = 0x2 (ASYNCMAP) value = 0x0 acked
22:21:08: PPP Async1: received config for type = 0x5 (MAGICNUMBER) value = 0x2A acked
22:21:08: PPP Async1: received config for type = 0x7 (PCOMPRESSION) acked
22:21:08: PPP Async1: received config for type = 0x8 (ACCOMPRESSION) acked
22:21:08: ipcp: sending CONFREQ, type = 3 (CI_ADDRESS), Address = 172.16.1.1
22:21:08: ppp Async1: ipcp_reqci: rcvd COMPRESSTYPE (rejected) (REJ)
22:21:08: ppp Async1: Negotiate IP address: her address 0.0.0.0 (NAK with address
172.16.1.100) (NAK)
22:21:08: ppp: ipcp_reqci: returning CONFREJ.
22:21:08: PPP Async1: state = REQSENT fsm_rconfack(0x8021): rcvd id 0x9
22:21:08: ipcp: config ACK received, type = 3 (CI_ADDRESS), Address = 172.16.1.1
22:21:08: ppp Async1: Negotiate IP address: her address 0.0.0.0 (NAK with address
172.16.1.100) (NAK)
22:21:08: ppp: ipcp_reqci: returning CONFNAK.
22:21:09: ppp Async1: Negotiate IP address: her address 172.16.1.100 (ACK)
22:21:09: ppp: ipcp_reqci: returning CONFACK.
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state to up
```

Verifying Autoselect ARA

The following trace appears when the **debug modem** and **debug arap internal** commands are enabled. As ARA version 2.0 calls pass through the access server, this output is displayed.

```
20:45:11: TTY3: DSR came up
20:45:11: tty3: Modem: IDLE->READY
20:45:11: TTY3: EXEC creation
20:45:11: TTY3: Autoselect(2) sample 1
20:45:11: TTY3: Autoselect(2) sample 11B
20:45:12: TTY3: Autoselect(2) sample 11B02
20:45:18: ARAP: ---------- SRVRVERSION ----------
20:45:19: ARAP: ---------- ACKing 0 ----------
20:45:19: ARAP: ---------- AUTH_CHALLENGE ----------
20:45:21: ARAP: ---------- ACKing 1 ----------
20:45:21: ARAP: ---------- AUTH_RESPONSE ----------
20:45:21: ARAP: ---------- STARTINFOFROMSERVER ----------
20:45:22: ARAP: ---------- ACKing 2 ----------
22:45:22: ARAP: ---------- ZONELISTINFO ----------
```

```
22:45:22: ARAP: ---------- ZONELISTINFO ----------
22:45:22: ARAP: ---------- ZONELISTINFO ----------
```

The following trace is for ARA version 1.0 calls:

```
22:31:45: TTY1: DSR came up
22:31:45: tty1: Modem: IDLE->READY
22:31:45: TTY1: Autoselect started
22:31:46: TTY1: Autoselect sample 16
22:31:46: TTY1: Autoselect sample 1610
22:31:46: TTY1: Autoselect sample 161002
22:31:47: ARAP: ---------- SRVRVERSION ----------
22:31:47: ARAP: ---------- ACKing 0 ----------
22:31:47: ARAP: ---------- AUTH_CHALLENGE ----------
22:31:47: ARAP: ---------- ACKing 1 ----------
22:31:47: ARAP: ---------- AUTH_RESPONSE ----------
22:31:47: ARAP: ---------- STARTINFOFROMSERVER ----------
22:31:48: ARAP: ---------- ACKing 2 ----------
22:31:48: ARAP: ---------- ZONELISTINFO ----------
22:31:48: ARAP: ---------- ZONELISTINFO ----------
22:31:49: ARAP: ---------- ZONELISTINFO ----------
```

How to Configure Other Asynchronous Line and Interface Features

This section describes the following asynchronous line and interface configurations:

- Configuring the Auxiliary (AUX) Port
- Establishing and Controlling the EXEC Process
- Enabling Routing on Asynchronous Interfaces
- Configuring Dedicated or Interactive PPP and SLIP Sessions
- Conserving Network Addresses
- Using Advanced Addressing Methods for Remote Devices
- Optimizing Available Bandwidth

Configuring the Auxiliary (AUX) Port

The AUX (auxiliary) port is typically configured as an asynchronous serial interface on routers without built-in asynchronous interfaces. To configure the AUX port as an asynchronous interface, configure it first as an auxiliary line with the **line aux 1** global configuration command.

The AUX port sends a data terminal ready (DTR) signal only when a Telnet connection is established. The auxiliary port does not use request to send/clear to send (RTS/CTS) handshaking for flow control. To understand the differences between standard asynchronous interfaces and AUX ports configured as an asynchronous interface, refer to Table 4. To enable the auxiliary port, use the following command in global configuration mode:

Command	Purpose
`Router(config)# line aux line-number`	Enables the auxiliary serial DTE port.

You cannot use the auxiliary (AUX) port as a second console port. To use the AUX port as a console port, you must order a special cable from your technical support personnel.

On an access server, you can configure any of the available asynchronous interfaces (1 through 8, 16, or 48). The auxiliary port (labeled AUX on the back of the product) can also be configured as an asynchronous serial interface, although performance on the AUX port is much slower than on standard asynchronous interfaces and the port does not support some features.

Table 4 illustrates why asynchronous interfaces permit substantially better performance than AUX ports configured as asynchronous interfaces.

Table 4 Differences Between the Asynchronous Port and the Auxiliary (AUX) Port

Feature	Asynchronous Interface	Auxiliary Port
Maximum speed	115200 bps	38400 bps
DMA buffering support[1]	Yes	No
PPP framing on chip[2]	Yes	No
IP fast switching[3]	Yes	No

1. Direct Memory Access (DMA) buffering moves data packets directly to and from system memory without interrupting the main CPU. This process removes overhead from the CPU and increases overall system performance.
2. PPP framing on a hardware chip removes overhead from the CPU on the router, which enables the router to sustain 115200 bps throughput on all asynchronous ports simultaneously.
3. After the destination of the first IP packet is added to the fast switching cache, it is fast switched to and from other interfaces with minimal involvement from the main processor.

On routers without built-in asynchronous interfaces, only the AUX port can be configured as an asynchronous serial interface. To configure the AUX port as an asynchronous interface, you must also configure it as an auxiliary line with the **line aux 1** command. Access servers do not have this restriction. Use the line command with the appropriate line configuration commands for modem control, such as speed.

Only IP packets can be sent across lines configured for SLIP. PPP supports transmission of IP, Internet Packet Exchange (IPX), and AppleTalk packets on an asynchronous serial interface.

See the "Line AUX Configuration Example" section for an example that shows how to configure the AUX port.

Establishing and Controlling the EXEC Process

By default, the Cisco IOS software starts an EXEC process on all lines. However, you can control EXEC processes, as follows:

- Turn the EXEC process on or off. (A serial printer, for example, should not have an EXEC session started.)
- Set the idle terminal timeout interval.

The EXEC command interpreter waits for a specified amount of time to receive user input. If no input is detected, the EXEC facility resumes the current connection. If no connections exist, it returns the terminal to the idle state and disconnects the incoming connection.

To control the EXEC process, use the following commands in line configuration mode:

	Command	Purpose
Step 1	`Router(config-line)# exec`	Turns on EXEC processes.
Step 2	`Router(config-line)# exec-timeout minutes [seconds]`	Sets the idle terminal timeout interval.

See the "High-Density Dial-In Solution Using Autoselect and EXEC Control Example" section for an example of configuring control over the EXEC process.

Enabling Routing on Asynchronous Interfaces

To route IP packets on an asynchronous interface, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# async dynamic routing`	Configures an asynchronous interface for dynamic routing. Use this command to manually bring up PPP from an EXEC session.
`Router(config-if)# async default routing`	Automatically configures an asynchronous interface for routing. Use this command to enable two routers to communicate over an asynchronous dial backup link.

The **async dynamic routing** command routes IP packets on an asynchronous interface, which permits you to enable the Interior Gateway Routing Protocol (IGRP), Routing Information Protocol (RIP), and Open Shortest Path First (OSPF) routing protocols for use when the user makes a connection using the **ppp** or **slip** EXEC commands. The user must, however, specify the **/routing** keyword at the SLIP or PPP command line.

For asynchronous interfaces in interactive mode, the **async default routing** command causes the **ppp** and **slip** EXEC commands to be interpreted as though the **/route** switch had been included in the command. For asynchronous interfaces in dedicated mode, the **async dynamic routing** command enables routing protocols to be used on the line. Without the **async default routing** command, there is no way to enable the use of routing protocols automatically on a dedicated asynchronous interface.

See the following sections for examples of enabling routing on asynchronous interfaces:

- Asynchronous Interface As the Only Network Interface Example
- IGRP Configuration Example

Configuring Dedicated or Interactive PPP and SLIP Sessions

You can configure one or more asynchronous interfaces on your access server (and one on a router) to be in dedicated network interface mode. In dedicated mode, an interface is automatically configured for SLIP or PPP connections. There is no user prompt or EXEC level, and no end-user commands are required to initiate remote-node connections. If you want a line to be used only for SLIP or PPP connections, configure the line for dedicated mode.

In interactive mode, a line can be used to make any type of connection, depending on the EXEC command entered by the user. For example, depending on its configuration, the line could be used for Telnet or XRemote connections, or SLIP or PPP encapsulation. The user is prompted for an EXEC command before a connection is initiated.

You can configure an asynchronous interface to be in dedicated network mode. When the interface is configured for dedicated mode, the end user cannot change the encapsulation method, address, or other parameters.

To configure an interface for dedicated network mode or to return it to interactive mode, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# async mode dedicated`	Places the line into dedicated asynchronous network mode.
`Router(config-if)# async mode interactive`	Returns the line to interactive mode.

By default, no asynchronous mode is configured. In this state, the line is not available for inbound networking because the SLIP and PPP connections are disabled.

See the "Dedicated Asynchronous Interface Configuration Example" section for an example of how to configure a dedicated asynchronous interface.

Conserving Network Addresses

When asynchronous routing is enabled, you might need to conserve network addresses by configuring the asynchronous interfaces as *unnumbered*. An unnumbered interface does not have an address. Network resources are therefore conserved because fewer network numbers are used and routing tables are smaller.

To configure an unnumbered interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip unnumbered type number`	Conserves IP addresses by configuring the asynchronous interfaces as unnumbered, and assigns the IP address of the interface type that you want to leverage.

Whenever the unnumbered interface generates a packet (for example, a routing update), it uses the address of the specified interface as the source address of the IP packet. It also uses the address of the specified interface to determine which routing processes are sending updates over the unnumbered interface.

You can use the IP unnumbered feature even if the system on the other end of the asynchronous link does not support it. The IP unnumbered feature is transparent to the other end of the link because each system bases its routing activities on information in the routing updates it receives and on its own interface address.

See the "Network Address Conservation Using the ip unnumbered Command Example" section for an example of how to conserve network addresses.

Using Advanced Addressing Methods for Remote Devices

You can control whether addressing is dynamic (the user specifies the address at the EXEC level when making the connection) or whether default addressing is used (the address is forced by the system). If you specify dynamic addressing, the router must be in interactive mode and the user will enter the address at the EXEC level.

It is common to configure an asynchronous interface to have a default address and to allow dynamic addressing. With this configuration, the choice between the default address or dynamic addressing is made by the users when they enter the **slip** or **ppp** EXEC command. If the user enters an address, it is used, and if the user enters the **default** keyword, the default address is used.

This section describes the following optional tasks:

- Assigning a Default Asynchronous Address
- Allowing an Asynchronous Address to Be Assigned Dynamically

Assigning a Default Asynchronous Address

To assign a permanent default asynchronous address, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# peer default ip address` *ip-address*	Assigns a default IP address to an asynchronous interface.

Use the **no** form of this command to disable the default address. If the server has been configured to authenticate asynchronous connections, you are prompted for a password after you enter the **slip default** or **ppp default** EXEC command before the line is placed into asynchronous mode.

The assigned default address is implemented when the user enters the **slip default** or **ppp default** EXEC command. The transaction is validated by the TACACS server, when enabled, and the line is put into network mode using the address that is in the configuration file.

Configuring a default address is useful when the user is not required to know the IP address to gain access to a system (for example, users of a server that is available to many students on a campus). Instead of each user being required to know an IP address, they only need to enter the **slip default** or **ppp default** EXEC command and let the server select the address to use.

See the section "Making Additional Remote Node Connections" in the chapter "Configuring Asynchronous SLIP and PPP" in this publication for more information about the **slip** and **ppp** EXEC commands.

See the following sections for examples:

- Modem Asynchronous Group Example
- Configuring Specific IP Addresses for an Interface
- IP and PPP Asynchronous Interface Configuration Example

Allowing an Asynchronous Address to Be Assigned Dynamically

When a line is configured for dynamic assignment of asynchronous addresses, the user enters the **slip** or **ppp** EXEC command and is prompted for an address or logical host name. The address is validated by TACACS, when enabled, and the line is assigned the given address and put into asynchronous mode.

Assigning asynchronous addresses dynamically is useful when you want to assign set addresses to users. For example, an application on a personal computer that automatically dials in using Serial Line Internet Protocol (SLIP) and polls for electronic mail messages can be set up to dial in periodically and enter the required IP address and password.

To assign asynchronous addresses dynamically, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# async dynamic address`	Allows the IP address to be assigned when the protocol is initiated.

The dynamic addressing features of the internetwork allow packets to get to their destination and back regardless of the access server, router, or network they are sent from. For example, if a host such as a laptop computer moves from place to place, it can keep the same address no matter where it is dialing in from.

Logical host names are first converted to uppercase and then sent to the TACACS server for authentication.

See the following sections for examples of configurations that allow asynchronous addresses to be assigned dynamically:

- Access Restriction on the Asynchronous Interface Example
- Asynchronous Routing and Dynamic Addressing Configuration Example
- Network Address Conservation Using the ip unnumbered Command Example

Optimizing Available Bandwidth

Asynchronous lines have relatively low bandwidth and can easily be overloaded, resulting in slow traffic across these lines.

To optimize available bandwidth, perform either of the following optional tasks:

- Configuring Header Compression
- Forcing Header Compression at the EXEC Level

Configuring Header Compression

One way to optimize available bandwidth is by using TCP header compression. Van Jacobson TCP header compression (defined by RFC 1144) can increase bandwidth availability two- to five-fold when compared to lines not using header compression. Theoretically, it can improve bandwidth availability by a ratio of seven to one.

To configure header compression, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip tcp header-compression [on | off | passive]`	Configures Van Jacobson TCP header compression on the asynchronous link.

Forcing Header Compression at the EXEC Level

On SLIP interfaces, you can force header compression at the EXEC prompt on a line on which header compression has been set to passive. This option allows more efficient use of the available bandwidth and does not require entering privileged configuration mode.

To implement header compression, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip tcp header-compression passive`	Allows status of header compression to be assigned at the user level.

For PPP interfaces, the **passive** option functions the same as the **on** option.

See the following sections for examples of header compression:

- TCP Header Compression Configuration Example
- Network Address Conservation Using the ip unnumbered Command Example
- IGRP Configuration Example

Configuration Examples for Asynchronous Interfaces and Lines

This section provides the following asynchronous interface configuration examples:

- Interface and Line Configuration Examples
- Line AUX Configuration Example
- Rotary Group Examples
- Dedicated Asynchronous Interface Configuration Example
- Access Restriction on the Asynchronous Interface Example
- Group and Member Asynchronous Interface Examples
- Asynchronous Interface Address Pool Examples
- IP and SLIP Using an Asynchronous Interface Example
- IP and PPP Asynchronous Interface Configuration Example
- Asynchronous Routing and Dynamic Addressing Configuration Example
- TCP Header Compression Configuration Example
- Network Address Conservation Using the ip unnumbered Command Example
- Asynchronous Interface As the Only Network Interface Example
- Routing on a Dedicated Dial-In Router Example
- IGRP Configuration Example

Interface and Line Configuration Examples

This section contains the following examples:

Asynchronous Interface Backup DDR Configuration Example

The following is an example of one asynchronous interface configuration on a Cisco AS2511-RJ access server that is used in an asynchronous backup DDR scenario:

```
interface async 1
 description ASYNC LINE 5293731 TO HIGHWAY
 encapsulation ppp
 async default routing
 async mode dedicated
 dialer in-band
 dialer map ip 192.168.10.2 name Router2 broadcast
 dialer-group 1
 ppp authentication chap
```

Passive Header Compression and Default Address Example

The following configuration shows interface and line configuration. The interface is configured with access lists, passive header compression, and a default address. The line is configured for TACACS authentication.

```
interface async 1
 ip access-group 1 in
 ip access-group 1 out
 ip tcp header-compression passive
 async default ip address 172.31.176.201

line 1
 login tacacs
 location 457-5xxx
 exec-timeout 20 0
 password XXXXXXXX
 session-timeout 20
 stopbits 1
```

High-Density Dial-In Solution Using Autoselect and EXEC Control Example

The following example configures a Cisco AS5800 access server, which is used as a high-density dial-in solution:

```
line 1/2/00 1/9/71
 session-timeout 30
 exec-timeout 30 0
 absolute-timeout 240
 autoselect during-login
 autoselect ppp
```

```
 modem InOut
 transport preferred none
 transport input all
```

Asynchronous Line Backup DDR Configuration Example

The following example configures one asynchronous line on a Cisco AS2511-RJ access server that is used in an asynchronous backup DDR scenario:

```
line 1
 modem InOut
 speed 115200
 transport input all
 flowcontrol hardware
```

Line AUX Configuration Example

In the following example, the asynchronous interface corresponds to the AUX port. Use the **show line** command to determine which asynchronous interface corresponds to the AUX port. The IP address on the AUX ports of both routers are in the same subnet

```
interface Async1
 ip address 192.168.10.1 255.255.255.0
 encapsulation ppp
 async dynamic routing
 async mode dedicated
!
no ip classless
ip route 0.0.0.0 0.0.0.0 Async1 /Default route points to the Async1 (AUX port) interface.
!
!
logging buffered
!
line con 0
 exec-timeout 0 0
line aux 0
 modem InOut
 transport input all
 rxspeed 38400
 txspeed 38400
```

Rotary Group Examples

The following example establishes a rotary group consisting of virtual terminal lines 2 through 4 and defines a password on those lines. By using Telnet to connect to TCP port 3001, the user gets the next free line in the rotary group. The user need not remember the range of line numbers associated with the password.

```
line vty 2 4
 rotary 1
 password letmein
 login
```

The following example enables asynchronous rotary line queueing:

```
line 1 2
 rotary 1 queued
```

The following example enables asynchronous rotary line queueing using the round-robin algorithm:

```
line 1 2
 rotary 1 queued round-robin
```

Dedicated Asynchronous Interface Configuration Example

The following example shows how to assign an IP address to an asynchronous interface and place the line in dedicated network mode. Setting the stop bit to 1 is a performance enhancement.

```
line 20
 location Department PC Lab
 stopbits 1
 speed 19200
!
interface async 20
 async default ip address 172.18.7.51
 async mode dedicated
```

Access Restriction on the Asynchronous Interface Example

The following example shows how to allow most terminal users access to anything on the local network, but restrict access to certain servers designated as asynchronous servers:

```
! access list for normal connections
access-list 1 permit 192.168.0.0 0.0.255.255
!
access-list 2 permit 192.168.42.55
access-list 2 permit 192.168.111.1
access-list 2 permit 192.168.55.99
!
line 1
 speed 19200
 flow hardware
 modem inout
interface async 1
 async mode interactive
 async dynamic address
 ip access-group 1 out
 ip access-group 2 in
```

Group and Member Asynchronous Interface Examples

The following examples are included in this section:

- Asynchronous Group Interface Examples
- Modem Asynchronous Group Example
- High-Density Dial-In Solution Using an Asynchronous Group

Asynchronous Group Interface Examples

The following example shows how to create an asynchronous group interface 0 with group interface members 2 through 7, beginning in global configuration mode:

```
interface group-async 0
 group-range 2 7
```

The following example shows how you need to configure asynchronous interfaces 1, 2, and 3 separately if you do not have a group interface configured:

```
interface Async1
 ip unnumbered Ethernet0
 encapsulation ppp
 async default ip address 172.30.1.1
 async mode interactive
 async dynamic routing
!
interface Async2
 ip unnumbered Ethernet0
 encapsulation ppp
 async default ip address 172.30.1.2
 async mode interactive
 async dynamic routing
!
interface Async3
 ip unnumbered Ethernet0
!
 encapsulation ppp
 async default ip address 172.30.1.3
 async mode interactive
 async dynamic routing
```

The following example configures the same interfaces, but from a single group asynchronous interface:

```
interface Group-Async 0
 ip unnumbered Ethernet0
 encapsulation ppp
 async mode interactive
 async dynamic routing
 group-range 1 3
 member 1 async default ip address 172.30.1.1
 member 2 async default ip address 172.30.1.2
 member 3 async default ip address 172.30.1.3
```

Modem Asynchronous Group Example

To configure a group asynchronous interface, specify the group async number (an arbitrary number) and the group range (beginning and ending asynchronous interface number).

The following example shows the process of creating and configuring a group asynchronous interface for asynchronous interfaces 1 through 96 on a Cisco AS5300 access server, which is loaded with ninety-six 56K MICA technologies modems:

```
interface group-async 1
 ip unnumbered ethernet 0
 encapsulation ppp
 async mode interactive
 ppp authentication chap pap
 peer default ip address pool default
 group-range 1 96
```

High-Density Dial-In Solution Using an Asynchronous Group

The following example configures a Cisco AS5800 access server that is used as a high-density dial-in solution:

```
interface group-async 0
 ip unnumbered FastEthernet0/2/0
 encapsulation ppp
 async mode interactive
 peer default ip address pool default
 no cdp enable
 ppp authentication chap
 hold-queue 10 in
 group-range 1/2/00 1/9/71
```

Asynchronous Interface Address Pool Examples

The following sections provide examples of the use of Dynamic Host Configuration Protocol (DHCP) and local pooling mechanisms:

- DHCP Pooling Example
- Local Pooling Example
- Configuring Specific IP Addresses for an Interface

DHCP Pooling Example

The following global configuration example enables DHCP proxy-client status on all asynchronous interfaces on the access server:

```
ip address-pool dhcp-proxy-client
```

The following global configuration example shows how to specify which DHCP servers are used on your network. You can specify up to four servers using IP addresses or names. If you do not specify servers, the default is to use the IP limited broadcast address of 255.255.255.255 for transactions with any and all discovered DHCP servers.

```
ip dhcp-server jones smith wesson
```

The following interface configuration example illustrates how to disable DHCP proxy-client functionality on asynchronous interface 1:

```
async interface
interface 1
 no peer default ip address
```

Local Pooling Example

The following example shows how to select the IP pooling mechanism and how to create a pool of local IP addresses that are used when a client dials in on an asynchronous line. The default address pool comprises IP addresses 172.30.0.1 through 172.30.0.28.

```
! This command tells the access server to use a local pool.
```

```
ip address-pool local
! This command defines the ip address pool.
! The address pool is named group1 and comprised of addresses.
! 172.30.0.1 through 172.30.0.28 inclusive
ip local-pool group1 172.30.0.1 172.30.0.28
```

Configuring Specific IP Addresses for an Interface

The following example shows how to configure the access server so that it will use the default address pool on all interfaces except interface 7, on which it will use an address pool called lass:

```
ip address-pool local
ip local-pool lass 172.30.0.1
 async interface
 interface 7
 peer default ip address lass
```

IP and SLIP Using an Asynchronous Interface Example

The following example configures IP and SLIP on asynchronous interface 6. The IP address for the interface is assigned to Ethernet 0, interactive mode has been enabled, and the IP address of the client PC running SLIP has been specified.

IP and the appropriate IP routing protocols have already been enabled on the access server or router.

```
interface async 6
 ip unnumbered ethernet 0
 encapsulation slip
 async mode interactive
 async default ip address 172.18.1.128
```

IP and PPP Asynchronous Interface Configuration Example

The following example configures IP and PPP on asynchronous interface 6. The IP address for the interface is assigned to Ethernet 0, interactive mode has been enabled, and the IP address of the client PC running PPP has been specified. IP and the appropriate IP routing protocols have already been enabled on the access server or router.

```
interface async 6
 ip unnumbered ethernet 0
 encapsulation ppp
 async mode interactive
 peer default ip address 172.18.1.128
```

Asynchronous Routing and Dynamic Addressing Configuration Example

The following example shows a simple configuration that allows routing and dynamic addressing.

With this configuration, if the user specifies **/routing** in the EXEC **slip** or **ppp** command, routing protocols will be sent and received.

```
interface async 6
 async dynamic routing
 async dynamic address
```

TCP Header Compression Configuration Example

The following example configures asynchronous interface 7 with a default IP address, allowing header compression if it is specified in the **slip** or **ppp** connection command entered by the user or if the connecting system sends compressed packets.

```
interface async 7
 ip address 172.31.79.1
 async default ip address 172.31.79.2
 ip tcp header-compression passive
```

Network Address Conservation Using the ip unnumbered Command Example

The following example shows how to configure your router for routing using unnumbered interfaces. The source (local) address is shared between the Ethernet 0 and asynchronous 6 interfaces (172.18.1.1). The default remote address is 172.18.1.2.

```
interface ethernet 0
 ip address 172.18.1.1 255.255.255.0
!
interface async 6
 ip unnumbered ethernet 0
 async dynamic routing
! Default address is on the local subnet.
 async dynamic address
 async default ip address 172.18.1.2
 ip tcp header-compression passive
```

The following example shows how the IP unnumbered configuration works. Although the user is assigned an address, the system response shows the interface as unnumbered, and the address entered by the user will be used only in response to BOOTP requests.

```
Router> slip /compressed 10.11.11.254
Password:
Entering async mode.
Interface IP address is unnumbered, MTU is 1500 bytes.
Header compression is On.
```

Asynchronous Interface As the Only Network Interface Example

The following example shows how one of the asynchronous lines can be used as the only network interface. The router is used primarily as a terminal server, but is at a remote location and dials in to the central site for its only network connection.

```
ip default-gateway 10.11.12.2
interface ethernet 0
 shutdown
interface async 1
 async dynamic routing
 ip tcp header-compression on
 async default ip address 10.11.16.12
 async mode dedicated
 ip address 10.11.12.32 255.255.255.0
```

Routing on a Dedicated Dial-In Router Example

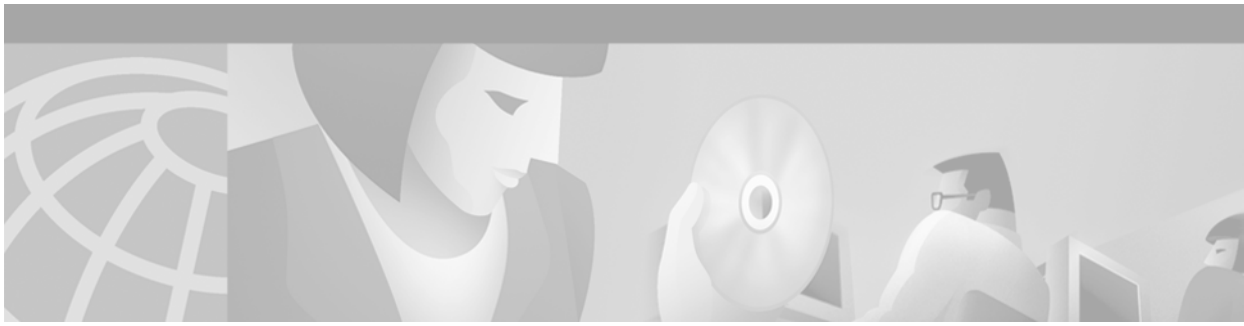
The following example shows how a router is set up as a dedicated dial-in router. Interfaces are configured as IP unnumbered to conserve network resources, primarily IP addresses.

```
ip routing
interface ethernet 0
 ip address 10.129.128.2 255.255.255.0
!
interface async 1
 ip unnumbered ethernet 0
 async dynamic routing
! The addresses assigned with SLIP or PPP EXEC commands are not used except
! to reply to BOOTP requests.
! Normally, the routers dialing in will have their own address and not use BOOTP at all.
 async default ip address 10.11.11.254
!
interface async 2
 ip unnumbered ethernet 0
 async default ip address 10.11.12.16
 ip tcp header-compression passive
 async mode dedicated
!
! Run RIP on the asynchronous lines because few implementations of SLIP
! understand IGRP. Run IGRP on the Ethernet (and in the local network).
!
router igrp 110
 network 10.11.12.0
! Send routes from the asynchronous lines on the production network.
 redistribute RIP
! Do not send IGRP updates on the asynchronous interfaces.
 passive-interface async 1
!
 router RIP
 network 10.11.12.0
 redistribute igrp
 passive-interface ethernet 0
! Consider filtering everything except a default route from the routing
! updates sent on the (slow) asynchronous lines.
 distribute-list 1 out
 ip unnumbered async 2
 async dynamic routing
```

IGRP Configuration Example

In the following example, only the Interior Gateway Routing Protocol (IGRP) TCP/IP routing protocol is running; it is assumed that the systems that are dialing in to use routing will either support IGRP or have some other method (for example, a static default route) of determining that the router is the best place to send most of its packets.

```
router igrp 111
 network 10.11.12.0
interface ethernet 0
 ip address 10.11.12.92 255.255.255.0
!
interface async 1
 async default ip address 10.11.12.96
 async dynamic routing
 ip tcp header-compression passive
 ip unnumbered ethernet 0

line 1
 modem ri-is-cd
```



Configuring Asynchronous Serial Traffic over UDP

This chapter describes how to communicate with a modem using the Asynchronous Serial Traffic over UDP feature in the following main sections:

- UDPTN Overview
- How to Configure Asynchronous Serial Traffic over UDP

See the "Configuration Examples for UDPTN" section for configuration examples.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the UDP commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

UDPTN Overview

The Asynchronous Serial Traffic over UDP feature provides the ability to encapsulate asynchronous data into User Datagram Protocol (UDP) packets and then unreliably send this data without needing to establish a connection with a receiving device. This process is referred to as UDP Telnet (UDPTN), although it does not—and cannot—use the Telnet protocol. UDPTN is similar to Telnet in that both are used to send data, but UDPTN is unique in that it does not require that a connection be established with a receiving device. You load the data that you want to send through an asynchronous port, and then send it, optionally, as a multicast or a broadcast. The receiving device(s) can then receive the data whenever it wants. If the receiver ends reception, the transmission is unaffected.

The Asynchronous Serial Traffic over UDP feature provides a low-bandwidth, low-maintenance method to unreliably deliver data. This delivery is similar to a radio broadcast: It does not require that you establish a connection to a destination; rather, it sends the data to whatever device wants to receive it. The receivers are free to begin or end their reception without interrupting the transmission.

It is a low-bandwidth solution for delivering streaming information for which lost packets are not critical. Such applications include stock quotes, news wires, console monitoring, and multiuser chat features.

This feature is particularly useful for broadcast, multicast, and unstable point-to-point connections. This feature may not work as expected when there are multiple users on the same port number in a nonmulticast environment. The same port must be used for both receiving and sending.

How to Configure Asynchronous Serial Traffic over UDP

To configure the Asynchronous Serial Traffic over UDP feature, perform the tasks described in the following sections:

- Preparing to Configure Asynchronous Serial Traffic over UDP (Required)
- Configuring a Line for UDPTN (Required)
- Enabling UDPTN (Required)
- Verifying UDPTN Traffic (Optional but Recommended)

See the "Configuration Examples for UDPTN" section at the end of this chapter for multicast, broadcast, and point-to-point UDPTN configuration examples.

Preparing to Configure Asynchronous Serial Traffic over UDP

When configuring the Asynchronous Serial Traffic over UDP feature for multicast transmission, you must configure IP multicast routing for the entire network that will receive or propagate the multicasts. When configuring the feature for broadcast transmission, you must configure broadcast flooding on the routers between network segments. Refer to the "Configuring IP Multicast Routing" chapter of this guide for information on how to configure IP multicast routing. See the section "Configuring Broadcast Packet Handling" in the *Cisco IOS IP Configuration Guide* for information on how to configure broadcast flooding.

Configuring a Line for UDPTN

To configure the line that will be used to send or receive UDP packets, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line line-number`	Enters line configuration mode for the line number specified.
Step 2	`Router(config-line)# transport output udptn`	Enables the line to transport UDP packets.
Step 3	`Router(config-line)# dispatch-timeout 1000`	Sends packets every 1000 milliseconds.
Step 4	`Router(config-line)# dispatch-character 13`	Sends packets after every new line.
Step 5	`Router(config-line)# no session-timeout`	Disables timeout connection closing.

Enabling UDPTN

There are two methods of enabling UDPTN. You can manually enable UDPTN when you want to begin transmission or reception, or you can configure the router to automatically enable UDPTN when a connection is made to the line.

To manually enable UDPTN and begin UDPTN transmission or reception, use the following command in EXEC mode:

Command	Purpose
`Router# udptn ip-address [port] [/transmit] [/receive]`	Enables UDPTN to the specified IP address (optionally, using the specified port). Use the **/transmit** or **/receive** keyword if the router will only be sending or receiving UDPTN.

To automatically enable UDPTN when a connection is made to the line, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line line-number`	Enters line configuration mode for the line number specified.
Step 2	`Router(config-line)# autocommand udptn ip-address [port] [/transmit] [/receive]`	Enables UDPTN automatically when a connection is made to the line (optionally, using the specified port). Use the **/transmit** or **/receive** keyword if the router will only be sending or receiving UDPTN.

Verifying UDPTN Traffic

To verify that UDPTN is enabled correctly, perform the following steps:

Step 1 Enable UDPTN debugging by using the **debug udptn** EXEC command.

Step 2 Enable UDPTN by using the **udptn** *ip-address* EXEC command, and then observe the debug output.

The following debug output shows a UDPTN session being successfully established and then disconnected.

```
Router# debug udptn
Router# udptn 172.16.1.1
Trying 172.16.1.1 ... Open

*Mar  1 00:10:15.191:udptn0:adding multicast group.
*Mar  1 00:10:15.195:udptn0:open to 172.16.1.1:57 Loopback0jjaassdd
*Mar  1 00:10:18.083:udptn0:output packet w 1 bytes
*Mar  1 00:10:18.087:udptn0:Input packet w 1 bytes
Router# disconnect
Closing connection to 172.16.1.1 [confirm] y
Router#
```

Step 3 While the **udptn** command is enabled, enter the **show ip socket** command to verify that the socket being used for UDPTN opened correctly.

```
Router# show ip socket
Proto    Remote      Port         Local       Port  In  Out  Stat TTY OutputIF
 17    --listen--            172.21.14.90      67   0    0     89      0
 17      0.0.0.0      520     172.21.14.90     520   0    0      1      0
 17      1.1.1.2       57        1.1.1.1       57   0    0     48      0
 17     224.1.1.1      57        1.2.2.2       57   0    0     48      0 Loopback0
```

Configuration Examples for UDPTN

This section provides the following UDPTN configuration examples:

- Multicast UDPTN Example
- Broadcast UDPTN Example
- Point-to-Point UDPTN Example

Multicast UDPTN Example

These configurations are for multicast UDPTN. The router that is multicasting does not require a multicast configuration—it simply sends to the multicast IP address.

Router That Is Multicasting

```
ip multicast-routing
interface ethernet 0
 ip address 10.1.1.1 255.255.255.0
 ip pim dense-mode
!
line 5
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 172.1.1.1 /transmit
```

Receiving Routers

```
ip multicast-routing
interface ethernet 0
 ip address 10.99.98.97 255.255.255.192
 ip pim dense-mode
!
line 0 16
 transport output udptn telnet lat rlogin
 autocommand udptn 172.1.1.1 /receive
```

Broadcast UDPTN Example

These configurations are for broadcast UDPTN. This is the simplest method to send to multiple receivers. The broadcasting router sends to the broadcast IP address, and any router that wants to receive the transmission simply connects to the broadcast IP address by using the **udptn** command.

Router That Is Broadcasting

```
interface ethernet 0
 ip address 10.1.1.1 255.255.255.0
!
line 5
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 255.255.255.255 /transmit
```

Receiving Routers

```
interface ethernet 0
 ip address 10.99.98.97 255.255.255.192
!
line 0 16
 transport output udptn telnet lat rlogin
 autocommand udptn 255.255.255.255 /receive
```

Point-to-Point UDPTN Example

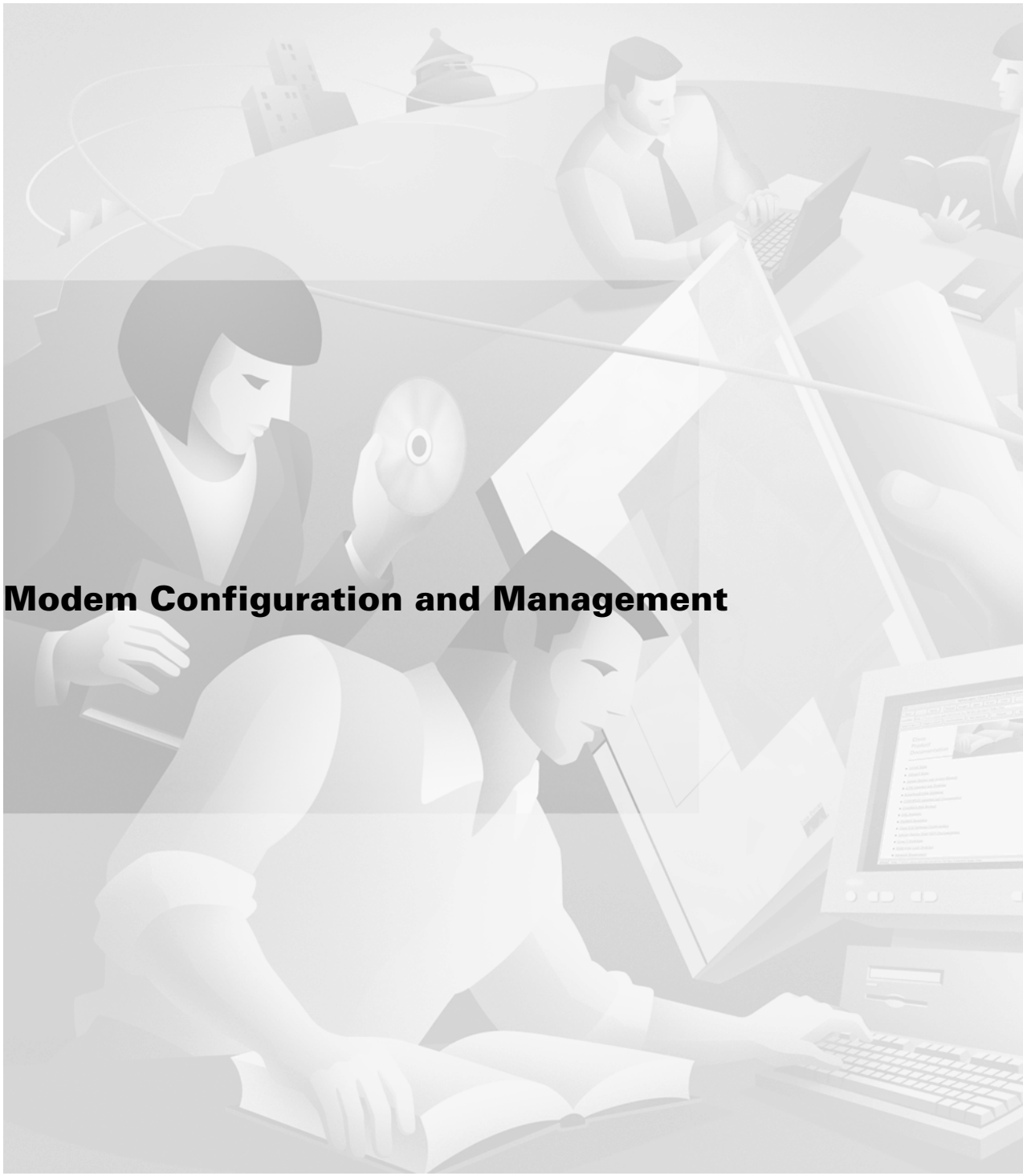
These configurations are for two routers in mobile, unstable environments that wish to establish a bidirectional asynchronous tunnel. Because there is no way to ensure that both routers will be up and running when one of the routers wants to establish a tunnel, they cannot use connection-dependent protocols like Telnet or local area transport (LAT). They instead use the following UDPTN configurations. Each router is configured to send to and receive from the IP address of the other. Because both routers will be sending and receiving, they do not use the **/transmit** or **/receive** keywords with the **udptn** command.

Router A

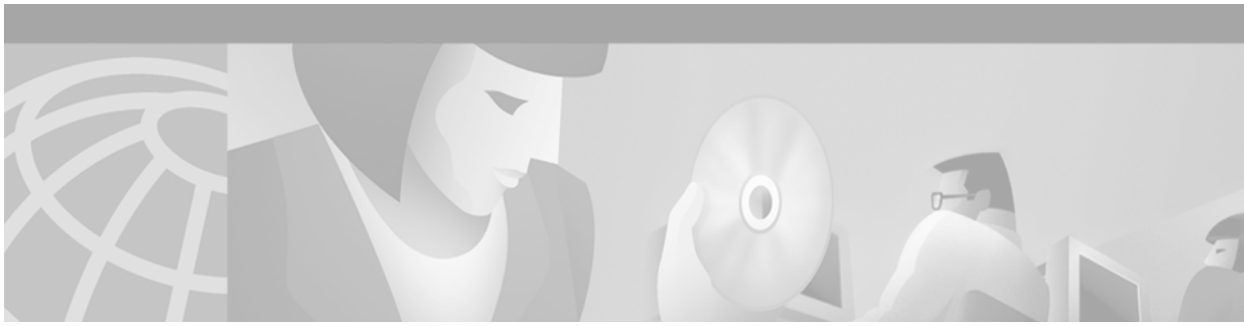
```
interface ethernet 0
 ip address 10.54.46.1 255.255.255.192
!
line 5
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 10.54.46.2
```

Router B

```
interface ethernet 0
 ip address 10.54.46.2 255.255.255.192
!
line 10
 no session-timeout
 transport output udptn
 dispatch-timeout 10000
 dispatch-character 13
 modem in
 autocommand udptn 10.54.46.1
```



Modem Configuration and Management



Overview of Modem Interfaces

This chapter describes modem interfaces in the following sections:

- Cisco Modems and Cisco IOS Modem Features
- Cisco IOS Modem Components
- Logical Constructs in Modem Configurations

See the chapter "Overview of Dial Interfaces, Controllers, and Lines" for more information about Cisco asynchronous serial interfaces.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Modem Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Cisco Modems and Cisco IOS Modem Features

Deciding which asynchronous features to use, to some degree, depends on your hardware configuration. All Cisco access servers must have their asynchronous interfaces and lines configured for network protocol support. Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration mode configure the physical and logical aspects for the same port.

Modems inside high-end access servers need a localized modem country code. This code is projected from the Cisco IOS software to the onboard modems using the **modem country** {**mica** | **microcom_hdms**} *country* command. The following are high-end access servers: Cisco AS5800, Cisco AccessPath, Cisco AS5300, and the Cisco AS5200.

Modems externally attached to low-end access servers need to receive initialization strings from the **modem autoconfigure discovery** command. For troubleshooting tips, see the section "External Modems on Low-End Access Servers" in the chapter "Configuring and Managing External Modems." The following are low-end access servers: Cisco AS2511-RJ, Cisco AS2509-RJ, Cisco 2509, Cisco 2511, and the Cisco 2512.

Figure 12 shows a Cisco AS2511-RJ access server. Figure 13 shows a Cisco AS5300 access server. Notice that modems are either inside or outside the chassis, depending on the product model.

Figure 12 Cisco AS2511-RJ Access Server

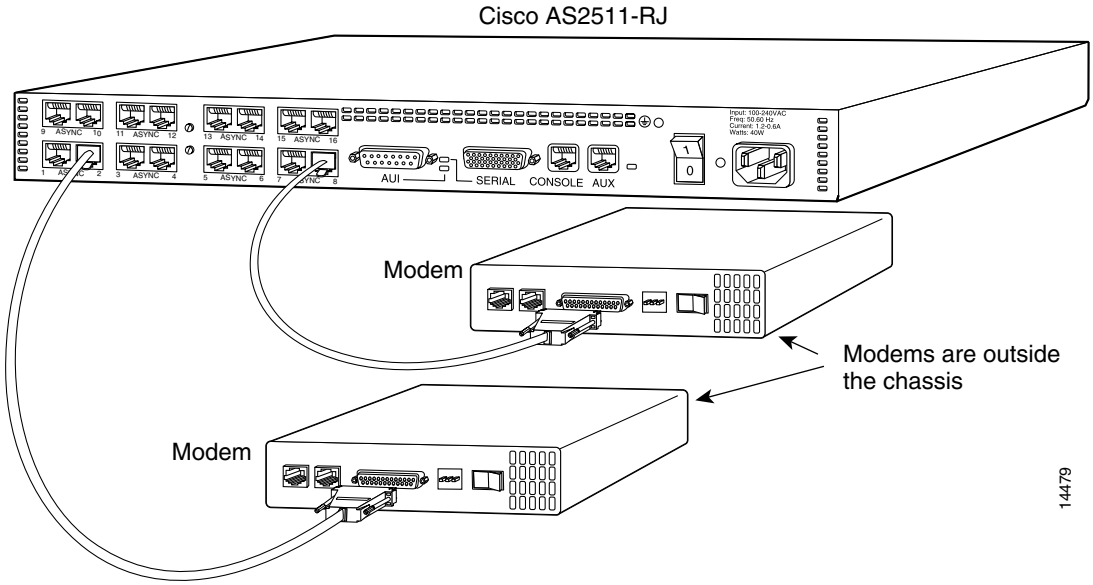
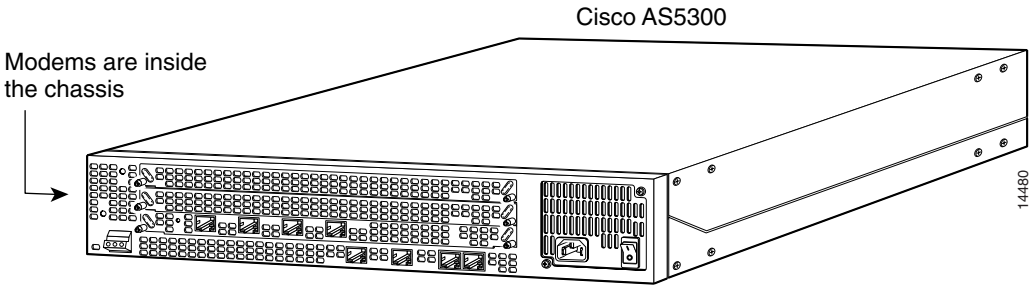


Figure 13 Cisco AS5300 Access Server

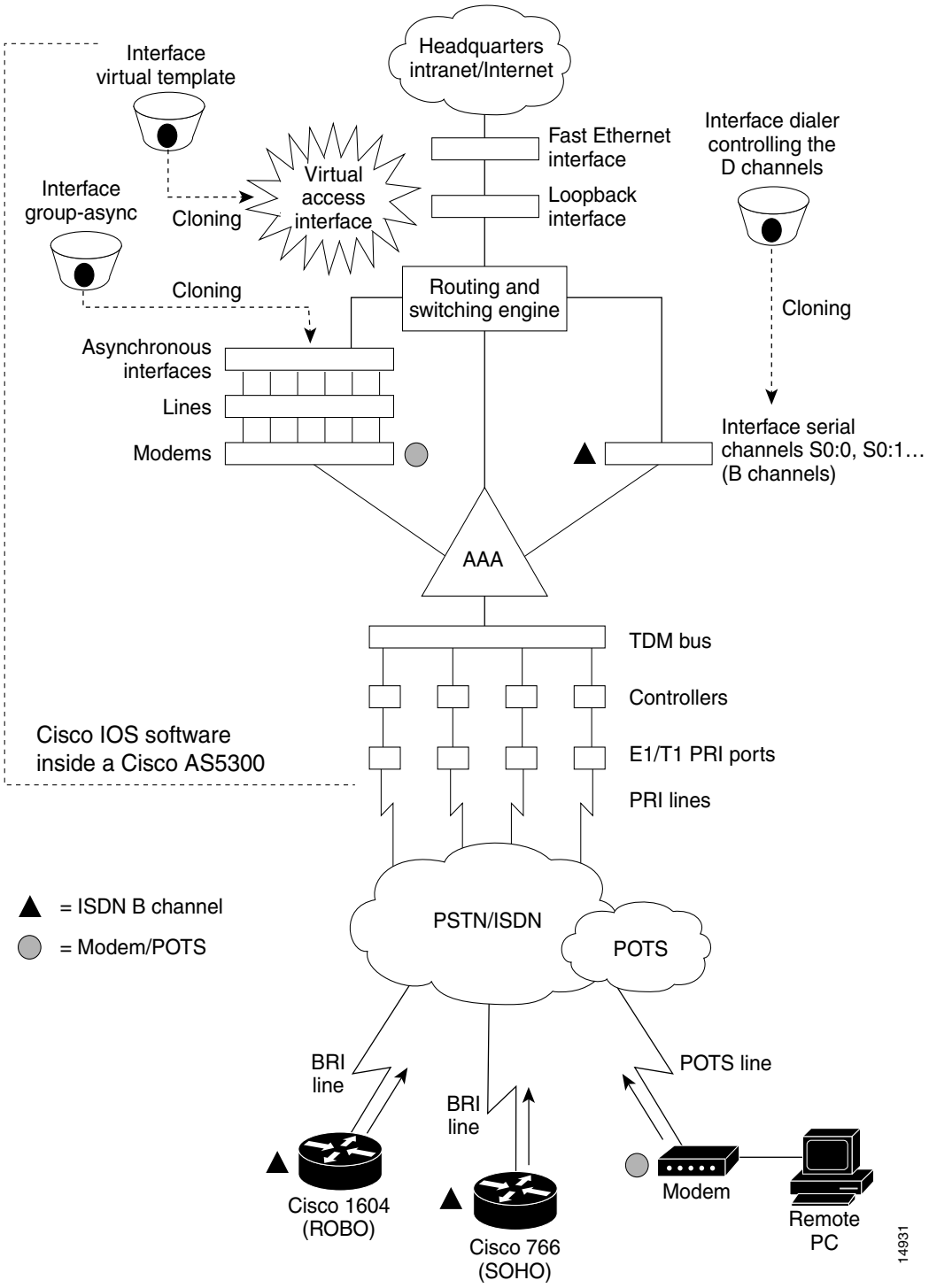


Cisco IOS Modem Components

Different components inside Cisco IOS software work together to enable remote clients to dial in and send packets. Figure 14 shows one Cisco AS5300 access server that is receiving calls from a remote office, branch office (ROBO); small office, home office (SOHO); and modem client.

Depending on your network scenario, you may encounter all of the components in Figure 14. For example, you might decide to create a virtual IP subnet by using a loopback interface. This step saves address space. Virtual subnets can exist inside devices that you advertise to your backbone. In turn, IP packets get relayed to remote PCs, which route back to the central site.

Figure 14 Cisco IOS Modem Concepts



Logical Constructs in Modem Configurations

A logical construct stores core protocol characteristics to assign to physical interfaces. No data packets are forwarded to a logical construct. Cisco uses three types of logical constructs in its access servers and routers. These constructs are described in the following sections:

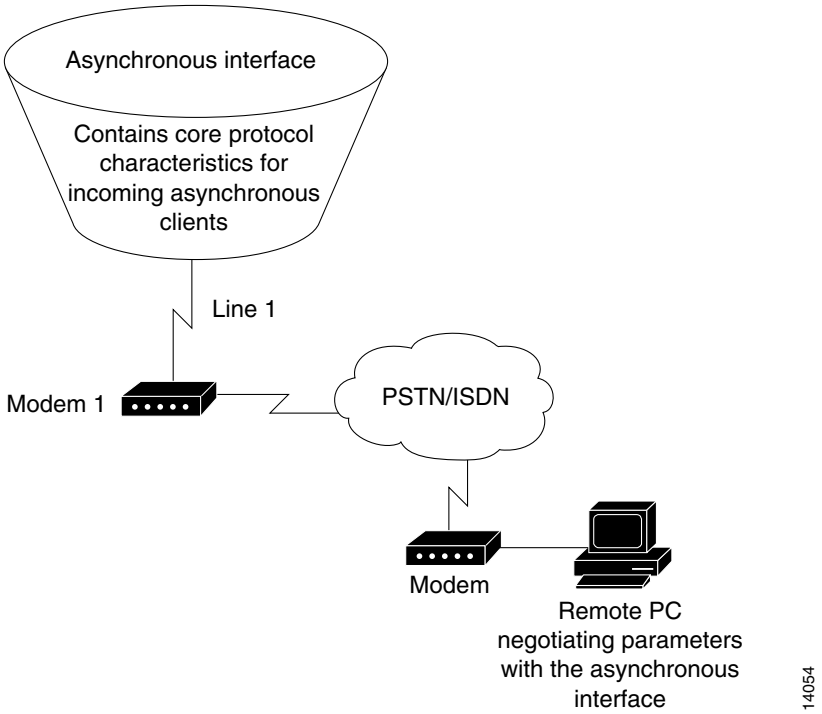
- Asynchronous Interfaces
- Group Asynchronous Interfaces
- Modem Lines and Asynchronous Interfaces

Asynchronous Interfaces

An asynchronous interface assigns network protocol characteristics to remote asynchronous clients that are dialing in through physical terminal lines and modems. (See Figure 15.)

Use the **interface async** command to create and configure an asynchronous interface.

Figure 15 Logical Construct for an Asynchronous Interface



To enable clients to dial in, you must configure two asynchronous components: asynchronous lines and asynchronous interfaces. Asynchronous interfaces correspond to physical terminal lines. For example, asynchronous interface 1 corresponds to tty line 1.

Commands entered in asynchronous interface mode configure protocol-specific parameters for asynchronous interfaces, whereas commands entered in line configuration mode configure the physical aspects for the same port.

Specifically, you configure asynchronous interfaces to support PPP connections. An asynchronous interface on an access server or router can be configured to support the following functions:

- Network protocol support such as IP, Internet Protocol Exchange (IPX), or AppleTalk
- Encapsulation support such as PPP
- IP client addressing options (default or dynamic)
- IPX network addressing options
- PPP authentication
- ISDN BRI and PRI configuration

For additional information about configuring asynchronous interfaces, see the "Overview of Dial Interfaces, Controllers, and Lines" chapter.

Group Asynchronous Interfaces

A group asynchronous interface is a parent interface that stores core protocol characteristics and projects them to a specified range of asynchronous interfaces. Asynchronous interfaces clone protocol information from group asynchronous interfaces. No data packets arrive in a group asynchronous interface.

By setting up a group asynchronous interface, you also eliminate the need to repeatedly configure identical configuration information across several asynchronous interfaces. For example, on a Cisco AS5300 one group asynchronous interface is used instead of 96 individual asynchronous interfaces. (See Figure 16.)

The following example shows a group asynchronous configuration for a Cisco AS5300 access server loaded with one 4-port ISDN PRI card and 96 MICA modems:

```
Router(config)# interface group-async 1
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# encapsulation ppp
Router(config-if)# async mode interactive
Router(config-if)# peer default ip address pool dialin_pool
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap pap dialin
Router(config-if)# group-range 1 96
```

To configure multiple asynchronous interfaces at the same time (with the same parameters), you can assign each asynchronous interface to a group and then configure the group. Configurations throughout this guide configure group asynchronous interfaces, rather than each interface separately.

If you want to configure different attributes on different asynchronous interfaces, do not assign them to the group or assign different interfaces to different groups. After assigning asynchronous interfaces to a group, you cannot configure these interfaces separately. For example, on a Cisco AS5300 access server in a T1 configuration, you could assign asynchronous interfaces 1 to 48 as part of one group (such as group-async1) and asynchronous interfaces 49 to 96 as part of another group (group-async2). You can also use the **member** command to perform a similar grouping function.

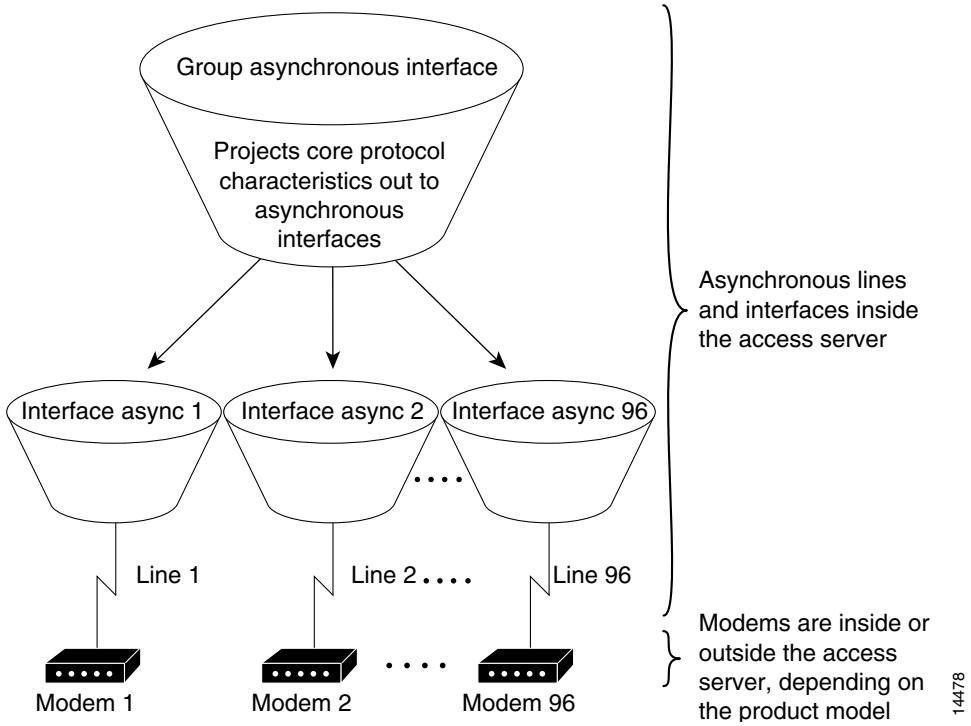
Modem Lines and Asynchronous Interfaces

Modems attach to asynchronous lines, which in turn attach to asynchronous interfaces. Depending on the type of access server you have, these components appear outside or inside the physical chassis. Figure 16 shows the logical relationships among modems, asynchronous lines, asynchronous interfaces, and group asynchronous interfaces. All these components work together to deliver packets as follows:

- Asynchronous calls come into the modems from the "plain old telephone service" (POTS) or Public Switched Telephone Network (PSTN).
- Modems pass packets up through asynchronous lines.
- Asynchronous interfaces clone their configuration information from group asynchronous interfaces.

Note The number of interfaces and modems varies among access server product models.

Figure 16 Modems, Lines, and Asynchronous Interfaces



Use the **interface group-async** command to create and configure a group asynchronous interface. The following example shows a group asynchronous configuration for a Cisco AS5300 access server loaded with one 4-port ISDN PRI card and 96 MICA modems:

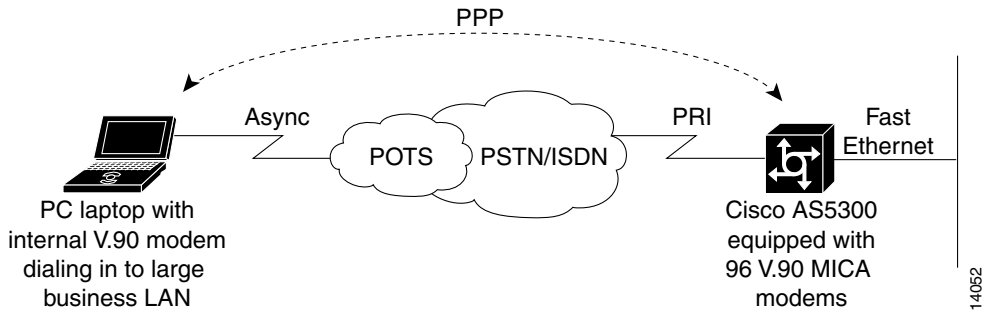
```
Router(config)# interface group-async 1
Router(config-if)# ip unnumbered loopback 0
Router(config-if)# encapsulation ppp
Router(config-if)# async mode interactive
Router(config-if)# peer default ip address pool dialin_pool
Router(config-if)# no cdp enable
Router(config-if)# ppp authentication chap pap dialin
Router(config-if)# group-range 1 96
```

Modem Calls

Modem calls travel through traditional telephone and ISDN lines. Regardless of the media used, these calls are initiated by a modem and terminate on another modem at the remote end.

Figure 17 shows a remote laptop using a V.90 internal modem to dial in to a Cisco AS5300 access server, which is loaded with 96 internal V.90 MICA technologies modems.

Figure 17 Remote Node Dialing In to a Cisco AS5300 Access Server



Asynchronous Line Configuration

Asynchronous line configuration commands configure ports for the following options:

- Physical layer options such as modem configuration
- Security for login in EXEC mode
- AppleTalk Remote Access (ARA) protocol configuration (PPP is configured in interface configuration mode)
- Autoselect to detect incoming protocols (ARA and PPP)

To enter line configuration mode, first connect to the console port of the access server and enter privileged EXEC mode. Then enter global configuration mode and finally enter line configuration mode for the asynchronous lines that you want to configure. The following example shows how you enter line configuration mode for lines 1 through 16:

```
Router> enable
Router# configure terminal
Router(config)# line 1 16
Router(config-line)#
```

Absolute Versus Relative Line Numbers

When you enter line configuration mode, you can specify an absolute line number or a relative line number. For example, absolute line number 20 is vty 2 (line 18 is vty 0). Referring to lines in a relative format is often easier than attempting to recall the absolute number of a line on a large system. Internally, the router uses absolute line numbers.

On all routers except the Cisco AS5350, AS5400, AS5800, AS5850 access servers, you can view all of the absolute and relative line numbers using the **show users all** EXEC command.

In the following sample display, absolute line numbers are listed at the far left. Relative line numbers are in the third column, after the line type. The second virtual terminal line, vty 1, is absolute line number 3. Compare the line numbers in this sample display to the output from the **show line** command.

```
Line       User     Host(s)               Idle Location
0 con 0
1 aux 0
2 vty 0             incoming              0 SERVER.COMPANY.COM
3 vty 1
4 vty 2
5 vty 3
6 vty 4
```

On the Cisco AS5350, AS5400, AS5800, AS5850 access servers, you can view the absolute and relative line numbers with the following commands:

- **show users all | exclude tty | interface** to show the non-internal modem lines
- **show controller async | include tty** to show the internal modem lines

The following example shows the information displayed with the **show users all | exclude tty|Interface** command:

```
Router# show users all | exclude tty | Interface
    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
   1 aux 0                                      00:00:00
   2 vty 0                                      00:00:00
   3 vty 1                                      00:00:00
   4 vty 2                                      00:00:00
   5 vty 3                                      00:00:00
   6 vty 4                                      00:00:00
```

The following example shows the information displayed with the **show controller async | include tty** command:

Router# show controller async | include tty

```
Controller information for Async2/00 (tty324)
Controller information for Async2/01 (tty325)
Controller information for Async2/02 (tty326)
.
.
.
```

Compare the line numbers in this sample display to the output from the **show line** command.

Line and Modem Numbering Issues

The tty line numbering scheme used by your access server or router is specific to your product and its hardware configuration. Refer to the product-specific documentation that came with your product for line numbering scheme information.

For example, the Cisco AS5200 access server has tty lines that map directly to integrated modems, as shown in Table 5. Depending on the shelf, slot, and port physical architecture of the access server, the modem and tty line number schemes will change.

As shown in Table 5, physical terminal lines 1 through 24 directly connect to modems 1/0 through 1/23, which are installed in the first chassis slot in this example. Physical terminal lines 25 through 48 directly connect to modems 2/0 through 2/23, which are installed in the second slot.

Table 5 tty Lines Associated with Cisco AS5200 Modems

tty Line	Slot/Modem Number	tty Line	Slot/Modem Number
1	1/0	25	2/0
2	1/1	26	2/1
3	1/2	27	2/2
4	1/3	28	2/3
5	1/4	29	2/4
6	1/5	30	2/5
7	1/6	31	2/6
8	1/7	32	2/7
9	1/8	33	2/8
10	1/9	34	2/9
11	1/10	35	2/10
12	1/11	36	2/11
13	1/12	37	2/12
14	1/13	38	2/13
15	1/14	39	2/14
16	1/15	40	2/15
17	1/16	41	2/16
18	1/17	42	2/17
19	1/18	43	2/18
20	1/19	44	2/19
21	1/20	45	2/20
22	1/21	46	2/21
23	1/22	47	2/22
24	1/23	48	2/23

Decimal TCP Port Numbers for Line Connections

Connections to an individual line are most useful when a dial-out modem, parallel printer, or serial printer is attached to that line. To connect to an individual line, the remote host or terminal must specify a particular TCP port on the router.

If reverse XRemote is required, the port is 9000 (decimal) plus the decimal value of the line number.

If a raw TCP stream is required, the port is 4000 (decimal) plus the decimal line number. The raw TCP stream is usually the required mode for sending data to a printer.

If Telnet protocols are required, the port is 2000 (decimal) plus the decimal value of the line number. The Telnet protocol might require that Return characters be translated into Return and line-feed character pairs. You can turn off this translation by specifying the Telnet binary mode option. To specify this option, connect to port 6000 (decimal) plus the decimal line number.

For example, a laser printer is attached to line 10 of a Cisco 2511 router. Such a printer usually uses XON/XOFF software flow control. Because the Cisco IOS software cannot receive an incoming connection if the line already has a process, you must ensure that an EXEC session is not accidentally started. You must, therefore, configure it as follows:

```
line 10
 flowcontrol software
 no exec
```

A host that wants to send data to the printer would connect to the router on TCP port 4008, send the data, and then close the connection. (Remember that line number 10 octal equals 8 decimal.)

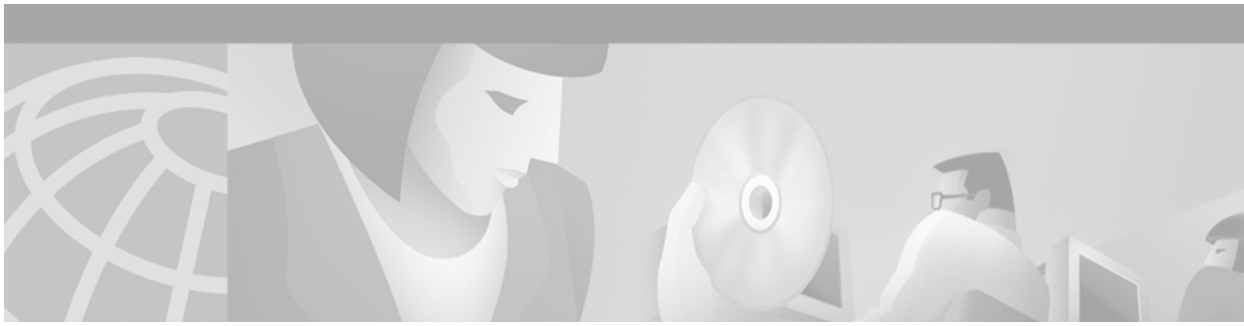
Signal and Flow Control Overview

The EIA/TIA-232 output signals are Transmit Data (TXDATA), Data Terminal Ready (DTR), and Ready To Send (RTS—Cisco 2500 routers only). The input signals are Receive Data (RXDATA), Clear to Send (CTS), and RING. The sixth signal is ground. Depending on the type of modem control your modem uses, these names may or may not correspond to the standard EIA/TIA-232 signals.

Dialup modems that operate over normal telephone lines at speeds of 28800 bps use hardware flow control to stop the data from reaching the host by toggling an EIA/TIA-232 signal when their limit is reached.

In addition to hardware flow control, modems require special software configuring. For example, they must be configured to create an EXEC session when a user dials in and to hang up when the user exits the EXEC. These modems also must be configured to close any existing network connections if the telephone line hangs up in the middle of a session.

The Cisco IOS software supports hardware flow control on its CTS input signal, which is also used by the normal modem handshake.



Configuring and Managing Integrated Modems

The Cisco IOS software provides commands that manage modems that reside inside access servers or routers in the form of modem cards. This chapter describes the modem management tasks. It includes the following main sections:

- Modems and Modem Feature Support
- Managing Modems
- Configuration Examples for Modem Management

For additional instructions for configuring Cisco access servers, see the chapter "Configuring and Managing Cisco Access Servers and Dial Shelves" in this publication.

To identify the hardware platform or software image information associated with a feature, 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.

Modem initialization strings are listed in the "Modem Initialization Strings" appendix. For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Modems and Modem Feature Support

The Cisco IOS software supports three types of integrated modems for Cisco access servers and access routers:

- Modem ISDN channel aggregation (MICA) digital modem
- NextPort digital modem
- NM-AM network module analog modem

Table 6 lists device support for each of the Cisco access server hardware platforms.

Table 6 Cisco IOS Modems and Modem Feature Support

Device Support	Cisco AS5300	Cisco AS5350	Cisco AS5400	Cisco AS5800	Cisco 2600/3600 Series Routers
Integrated modems	6- and 12-port MICA	60-port NextPort CSM v6DFC	108-port NextPort CSM v6DFC	72- and 144-port MICA 324-port NextPort CSM v6DFC	6-port, 12-port, 18-port, 24-port, or 30-port MICA NM-DM 8- and 16-port analog NM-AM
V.90	Yes	Yes	Yes	Yes	Yes with NM-DM
V.110	Yes	Yes	Yes	Yes	Yes with NM-DM
V.120	No, CPU only	Yes	Yes	Yes with 324-port NextPort[1] CSM v6DFC	No, CPU only

1. For more detailed information regarding the V.120 functionalities that are supported both by NextPort and Cisco IOS software, see the section "V.120 Bit Rate Adaptation Standard."

Note If the platform is using MICA technologies modems, the V.120 rate adaptation is done by CPU on vty lines like protocol translation sessions.

The following sections summarize the standards supported by modems in the Cisco access servers. See Table 7 through Table 10 for a summary and comparison of the Cisco IOS commands used for the MICA and NextPort modems.

V.90 Modem Standard

Study Group 16 of the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) developed the V.90 modem standard for multimedia systems. The V.90 standard describes a digital modem and analog modem pair for use on the public switched telephone network (PSTN). V.90 modems are designed for connections that are digital at one end and have only one digital-to-analog conversion. The V.90 standard is expected to be widely used for applications such as Internet and online service access. Download speeds of up to 56,000 bits per second (bps) are possible, depending on telephone line conditions, with upload speeds of up to 33,600 bps.

V.110 Bit Rate Adaption Standard

V.110 is a bit rate adaptation standard defined by the ITU that provides a standard method of encapsulating data over global system for mobile telecommunication (GSM) and ISDN networks. V.110 allows for reliable transport of asynchronous or synchronous data. V.110 adapts a low-speed connection

to an ISDN B channel allowing the remote station or terminal adapter to use the fast call setup times offered by ISDN. This feature allows V.110 calls to be originated and terminated over ISDN. It also enables GSM wireless connectivity.

V.110, as an alternative to V.120, provides DTE with V-series type interfaces with access to ISDN network by bit stuffing. Many V.110 devices are used in Europe and Japan. In Japan, MICA supports the Personal-Handyphone-System Internet Access Forum Standard (PIAFS) protocol, which is similar to V.110.

The V.110 implementation for calls on MICA modems is managed by special boardware and modem code, along with the appropriate Cisco IOS image, in a manner similar to other modulation standards. This MICA V.110 implementation provides V.110 user rates ranging from 600 bps to 38,400 bps.

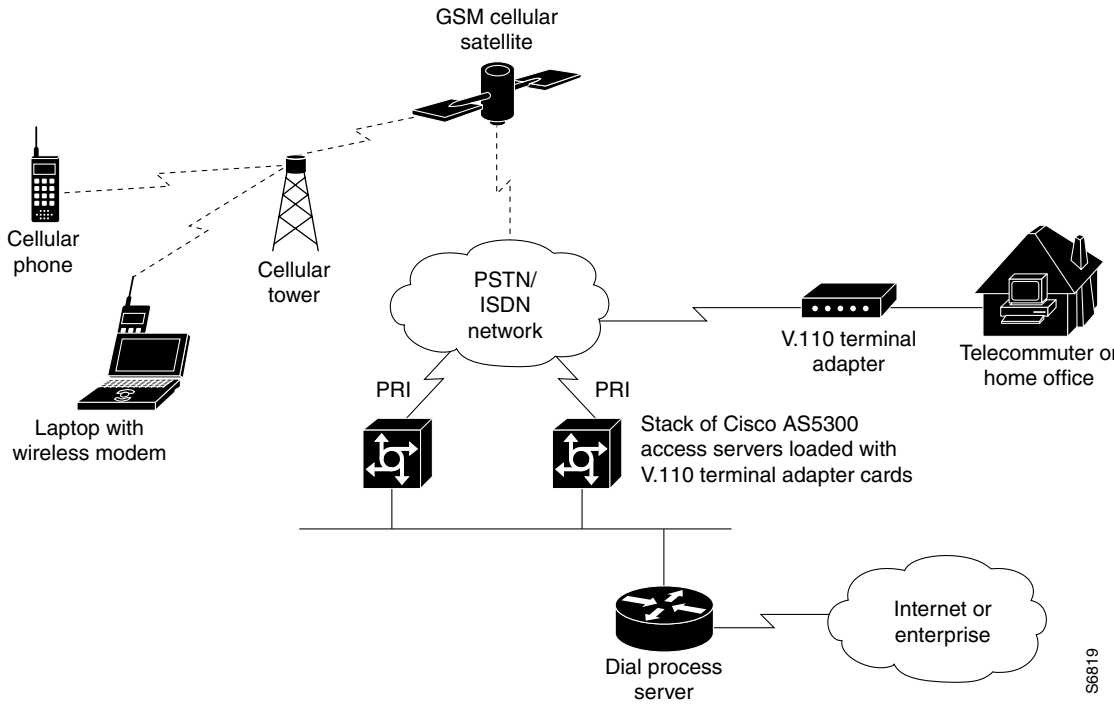
V.110 is supported on the following Cisco devices and network modules:

- Cisco AS5300-series access servers
- Cisco 3620, 3640, and 3660 access routers
- NM-6DM, NM-12DM, NM-18DM, NM-24DM, and NM-30DM network modules

The digital signal processors (DSPs) on the board can function as either modems or V.110 terminal adapters (or V.120 terminal adapters for NextPort DSPs). Based on the ISDN Q.931 bearer capability information element, the Cisco IOS software configures the DSP to treat the incoming call as a modem call, a V.110 call, or a V.120 call.

Figure 18 shows a dial-in scenario for how V.110 technology can be used with a stack of Cisco AS5300-series access servers.

Figure 18 V.110 Dial-In Scenario Using a Stack of Cisco AS5300-Series Access Servers



V.120 Bit Rate Adaptation Standard

ITU-T Recommendation V.120 revised by the ITU-T Study Group 14. V.120 describes a standard that can be used for adapting terminals with non-ISDN standard network interfaces to an ISDN. It is intended to be used between two terminal adapter (TA) functional groups, between two ISDN terminal (TE1) functional groups, between a TA and a TE1, or between either a TA or TE1 and an interworking facility inside a public or private ISDN.

V.120 allows for reliable transport of synchronous, asynchronous, or bit transparent data over ISDN bearer channels. Cisco provides three V.120 support features for terminal adapters that do not send the low-layer compatibility fields or bearer capability V.120 information:

- Answer all incoming calls as V.120—Static configuration used when all remote users have asynchronous terminals and need to connect with a vty on the router.
- Automatically detect V.120 encapsulation—Encapsulation dynamically detected and set.
- Enable V.120 support for asynchronous access over ISDN.

For terminal adapters that send the low-layer compatibility or bearer capability V.120 information, mixed V.120 and ISDN calls are supported. No special configuration is required.

V.120 is a digital rate adaptation and cannot be done on NM-AM network module analog modems. MICA DSP firmware does not have the code to terminate V.120 calls.

NextPort supports only a subset of V.120 functionalities that are supported by Cisco IOS software. Therefore, certain V.120 calls still will need to be terminated on the CPU, even if the chassis has available NextPort modems.

Managing Modems

To manage modems, perform the tasks in the following sections; the tasks you need to perform depend upon the type and needs of your system:

- Managing SPE Firmware
- Configuring Modems in Cisco Access Servers
- Configuring Cisco Integrated Modems Using Modem Attention Commands
- Configuring Modem Pooling
- Configuring Physical Partitioning
- Configuring Virtual Partitioning
- Configuring Call Tracker
- Configuring Polling of Link Statistics on MICA Modems
- Configuring MICA In-Band Framing Mode Control Messages
- Enabling Modem Polling
- Setting Modem Poll Intervals
- Setting Modem Poll Retry
- Collecting Modem Statistics
- Troubleshooting Using a Back-to-Back Modem Test Procedure
- Clearing a Direct Connect Session on a Microcom Modem

Managing SPE Firmware

You can upgrade your modem firmware to the latest NextPort Service Processing Element (SPE) firmware image available from Cisco. The SPE firmware image is usually retrieved from Cisco.com. You must first copy the SPE image from a TFTP server to flash memory using the **copy tftp flash** command. You then configure the firmware upgrade using the **firmware location** and **firmware upgrade** SPE configuration commands. The **firmware location** command specifies the location of the firmware file and downloads the firmware to an SPE or a range of SPEs, according to the schedule you selected for the firmware upgrade method using the **firmware upgrade** command.

The modem firmware upgrade commands must be saved into the system configuration using the **write memory** command; otherwise, at the next reboot downloading of the specified firmware will not occur.

To upgrade SPE firmware, use the following commands:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.

Step 2	AS5400:
	`Router(config)# spe slot/spe`
	or
	`Router(config)# spe slot/spe slot/spe`
	
	AS5800:
	`Router(config)# spe shelf/slot/spe`
	or
	`Router(config)# spe shelf/slot/spe shelf/slot/spe`
		Enters SPE configuration mode. You can choose to configure a range of SPEs by specifying the first and last SPE in the range.

Step 3	`Router(config-spe)# firmware upgrade {busyout | download-maintenance | reboot}`	Specifies the upgrade method. Three methods of upgrade are available. The **busyout** keyword waits until all calls are terminated on an SPE before upgrading the SPE to the designated firmware. The **download-maintenance** keyword upgrades the firmware during the download maintenance time. The **reboot** keyword requests the access server to upgrade firmware at the next reboot.

	Command	Purpose
Step 4	`Router(config-spe)# firmware location [IFS:[/]]filename`	Specifies the SPE firmware file in flash memory to use for the selected SPEs. Allows you to upgrade firmware for SPEs after the new SPE firmware image is copied to your flash memory. The Cisco IOS file specification (IFS) can be any valid IFS on any local file system. Use the **dir all-filesystems** EXEC command to display legal IFSs. Examples of legal IFS specifications include: • **bootflash:**—Loads the firmware from a separate flash memory device. • **flash:**—Loads the firmware from the flash NVRAM located within the router. • **system:/**—Loads the firmware from a built-in file within the Cisco IOS image. The optional forward slash (/) and system path must be entered with this specification. • *filename*—The name of the desired firmware file (for example, mica-modem-pw.2.7.3.0.bin). If the **system** keyword is specified, enter the path to the filename you want to download.
Step 5	`Router(config-spe)# exit`	Exits SPE configuration mode.
Step 6	`Router(config)# exit`	Exits global configuration mode.
Step 7	`Router# copy running-config startup-config`	Saves your changes.

Note As soon as a firmware file is specified, the downloading begins. Do not specify all modems and then go into an upgrade process on a busy router. The modems that are not busy will all be marked busy and the server will wait until all the modems on each of the given cards are free before upgrading the multiple-port cards. The only way to clear this situation is to start disconnecting users with a **clear** command. Normally, groups of modems are specified in scripts with the spe slot/spe_begin and slot/spe_end statements, and upgrades are done in a rolling fashion.

Use the **show modem version** and **show spe version** commands to verify that the modems are running the portware version you specified.

The following example shows how to enter the SPE configuration mode, set the range of SPEs, specify the firmware file location in flash memory, download the file to the SPEs, and display a status report using the **show spe** EXEC command:

```
Router# configure terminal
Router(config)# spe 7/0 7/17
Router(config-spe)# firmware upgrade busyout
Router(config-spe)# firmware location flash:np_6_75
Started downloading firmware flash:np_6_75.spe
Router(config-spe)# exit
Router(config)# exit
Router# show spe 7
.
.
.
```

```
                        SPE             SPE      SPE  SPE    Port          Call
SPE#     Port #         State           Busyout  Shut Crash  State         Type
7/00     0000-0005      ACTIVE                1     0     0  BBBBBB        _______
7/01     0006-0011      DOWNLOAD              1     0     0  bbbbbb        _______
7/02     0012-0017      DOWNLOAD              1     0     0  bbbbbb        _______
7/03     0018-0023      DOWNLOAD              1     0     0  bbbbbb        _______
.
.
.
```

For information about upgrading Cisco 3600 Series and Cisco 3700 modems, see the *Cisco 3600 Series and Cisco 3700 Series Modem Portware Upgrade Configuration Note* at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/sw_conf/portware/5257d56k.htm .

Configuring Modems in Cisco Access Servers

To configure modem support for access servers such as the Cisco AS5300 and AS5800, perform the following tasks. The list describes which tasks are required and which are optional but recommended.

- Configuring Modem Lines (Required)
- Verifying the Dial-In Connection (Optional but Recommended)
- Troubleshooting the Dial-In Connection (Optional but Recommended)
- Configuring the Modem Using a Modemcap (Required)
- Configuring the Modem Circuit Interface (Required for Digital Modems)

Note See the chapter "Configuring and Managing Cisco Access Servers and Dial Shelves" for additional information about configuring Cisco AS5x00 series access servers.

Configuring Modem Lines

You must configure the modem lines and set the country code to enable asynchronous connections into your access server. To configure the modems and line, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**MICA modems**
	`Router(config)# modem country mica country`
	NextPort SPE modems
	`Router(config)# spe country country`
	Microcom modems
	`Router(config)# modem country microcom_hdms country`
		Depending on the type of modems loaded in your access server, specifies the modem vendor and country code.[1] This step is only for the MICA, NextPort SPE, and Microcom modems in the Cisco AS5000 series access servers.
		Table 7 through Table 10 provide a summary and comparison of the Cisco IOS commands used for the MICA and NextPort modems.

Step 2	`Router(config)# line beginning-line-number ending-line-number`	Enters the number of modem lines to configure. Usually this range is equal to the number of modems in the access server. Use the **show line** EXEC command to see which lines are available.

	Command	Purpose
Step 3	`Router(config-line)# transport {input | output} {all | none}`	Specifies that connection protocols can be used when connecting to the line. For outgoing calls, choose the **output** option. For incoming calls, choose the **input** option. If you do not intend to dial out, choose the **none** option.
Step 4	`Router(config-line)# autoselect {arap | ppp | slip}`	Configures the line to automatically startup an AppleTalk Remote Access (ARA), PPP, and Serial Line Internet Protocol (SLIP) session. You can configure more than one protocol by entering multiple **autoselect** commands with the appropriate keyword.
Step 5	`Router(config-line)# autoselect during-login`	Configures the lines to display the username and password prompt as soon as the line is connected, rather than waiting until the user presses the Enter or Return key at the terminal.
Step 6	`Router(config-line)# login authentication dialin` or `Router(config-line)# login` *login-name* `Router(config-line)# password` *password*	Enables authentication across all asynchronous modem logins. Use the **login authentication dialin** command when authentication, authorization, and accounting (AAA) authentication has been enabled. Use the **login** and **password** commands to configure non-AAA user authentication.
Step 7	`Router(config-line)# modem dialin`	Configures the modem for only incoming calls.
Step 8	`Router(config-line)# exit`	Returns to global configuration mode.

1. For a comprehensive list of modem country codes, see the **modem country mica** command and the **modem country microcom_hdms** command in the *Cisco IOS Dial Technologies Command Reference*.

Verifying the Dial-In Connection

Before configuring any additional protocols for the line such as SLIP, PPP, or ARA, test whether the dial-in connection for the access server and modem are configured correctly for dial-in access,

Note The same configuration issues exist between the client DTE and client modem. Make sure that you have the correct EIA/TIA-232 cabling and modem initialization string for your client modem.

The following is an example of a successful connection from a PC using a known good modem to dial in to a Cisco access server:

```
at
OK
atdt9,5550101
CONNECT 14400/ARQ/V32/LAPM/V42BIS
User Access Verification
Username: user1
Password:
Router>
```

Troubleshooting the Dial-In Connection

Depending upon the problems you experience, take the appropriate action:

- If you are having problems making or receiving calls, make sure that you turned on the protocols for connecting to the lines and configured for incoming and outgoing calls.
- If the calls are not coming up at all, turn on modem debugging. Use the the modem debugging commands as follows:
  - The **debug modem** command enables debugging on the modem line.
  - The **debug modem csm** (or **debug csm modem**) command enables debugging for lines configured for digital modems.
  - The **debug isdn q931** command enables debugging for lines configured for the ISDN and Signaling System 7 (SS7) Q.931 protocols.
  - The **debug cas** command enables debugging for lines configured for channel-associated signaling (CAS).

  Following is a sample of how to enable and then disable Cisco IOS modem debugging commands on a network access server:

```
Router# debug modem
Router# debug modem csm
Router# debug isdn q931
Router# no debug modem
Router# no debug modem csm
Router# no debug isdn q931
```

- Enter the **debug modem ?** command for a list of additional modem debugging commands:

```
Router# debug modem ?
  b2b          Modem Special B2B
  csm          CSM activity
  maintenance  Modem maintenance activity
  mica         MICA Async driver debugging
  oob          Modem out of band activity
  tdm          B2B Modem/PRI TDM
  trace        Call Trace Upload
```

- Turn off the messages by entering the **no debug modem** command.

For more detailed information refer to the TAC Tech Notes document, *Troubleshooting Modems*, at the following URL: http://www.cisco.com/warp/public/471/index_14280.html

Configuring the Modem Using a Modemcap

Modems are controlled by a series of parameter settings (up to a limit of 128 characters) that are sent to the modem to configure it to interact with a Cisco device in a specified way. The parameter settings are stored in a database called a *modem capability* (modemcap). The Cisco IOS software contains defined modemcaps that have been found to properly initialize internal modems. Following are the names of some modemcaps available in the Cisco IOS software:

- cisco_v110—Cisco (NEC) internal V.110 TA (AS5200)
- mica—Cisco MICA HMM/DMM internal digital modem
- nextport—Cisco NextPort CSMV/6 internal digital modem
- microcom_hdms—Microcom HDMS chassis

- microcom_mimic—Cisco (Microcom) internal analog modem (NM-AM–2600/3600)
- microcom_server—Cisco (Microcom) V.34/56K internal digital modem (AS5200)

Enter these modemcap names with the **modem autoconfigure type** command.

For more information on creating and using modemcaps refer to the TAC Tech Notes documentation, *Recommended Modemcaps for Internal Digital and Analog Modems on Cisco Access Servers*, at the following URL: http://www.cisco.com/warp/public/471/recc_modemcaps.html

If your modem is not on this list and if you know what modem initialization string you need to use with it, you can create your own modemcap; see the following procedure, "Using the Modem Autoconfigure Type Modemcap Feature." To have the Cisco IOS determine what type of modem you have, use the **modem autoconfigure discovery** command to configure it, as described in the procedure "Using the Modem Autoconfigure Discovery Feature."

Note When configuring an internal modem, avoid using the Modem Autoconfigure Discovery feature because the feature can misdetect the internal modem type and cause the modem to start working in an unpredictable and unreproducable manner.

Using the Modem Autoconfigure Type Modemcap Feature

If you know what modem initialization string you need to use with your modem, you can create your own modemcap by performing the following steps.

Step 1 Use the **modemcap edit** command to define your own modemcap entry.

The following example defines modemcap MODEMCAPNAME:

```
Router(config)# modemcap edit MODEMCAPNAME miscellaneous &FS0=1&D3
```

Step 2 Apply the modemcap to the modem lines as shown in the following example:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#line 33 34
Router(config-line)#modem autoconfigure type MODEMCAPNAME
Jan 16 18:12:59.643: TTY34: detection speed (115200) response ---OK---
Jan 16 18:12:59.643: TTY34: Modem command:  --AT&FS0=1&D3--
Jan 16 18:12:59.659: TTY33: detection speed (115200) response ---OK---
Jan 16 18:12:59.659: TTY33: Modem command:  --AT&FS0=1&D3--
Jan 16 18:13:00.227: TTY34: Modem configuration succeeded
Jan 16 18:13:00.227: TTY34: Detected modem speed 115200
Jan 16 18:13:00.227: TTY34: Done with modem configuration
Jan 16 18:13:00.259: TTY33: Modem configuration succeeded
Jan 16 18:13:00.259: TTY33: Detected modem speed 115200
Jan 16 18:13:00.259: TTY33: Done with modem configuration
```

Note The report that is generated by the **debug confmodem** command can be misleading for the MICA and NextPort internal modems because these modems do not have Universal Asynchronous Receiver/Transmitter (UART) and exchange data with the CPU at speeds of hundreds of kbps.

Using the Modem Autoconfigure Discovery Feature

If you prefer that the modem software use its autoconfigure mechanism to configure the modem, use the **modem autoconfigure discovery** command.

The following example shows how to configure modem autoconfigure discovery mode:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# line 33 34
Router(config-line)# modem autoconfigure discovery
Jan 16 18:16:17.724: TTY33: detection speed (115200) response ---OK---
Jan 16 18:16:17.724: TTY33: Modem type is default
Jan 16 18:16:17.724: TTY33: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:17.728: TTY34: detection speed (115200) response ---OK---
Jan 16 18:16:17.728: TTY34: Modem type is default
Jan 16 18:16:17.728: TTY34: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:18.324: TTY33: Modem configuration succeeded
Jan 16 18:16:18.324: TTY33: Detected modem speed 115200
Jan 16 18:16:18.324: TTY33: Done with modem configuration
Jan 16 18:16:18.324: TTY34: Modem configuration succeeded
Jan 16 18:16:18.324: TTY34: Detected modem speed 115200
Jan 16 18:16:18.324: TTY34: Done with modem configuration
```

Configuring the Modem Circuit Interface

The next task to complete before using the integrated modem is to configure the modem circuit interface. The basic steps are outlined next:

- If the integrated modem is an analog modem, no further configuration is required; modem characteristics are set on the line.
- If the integrated modem is a digital modem, you can configure either the ISDN or CAS, as appropriate.
  - For ISDN BRI and PRI, you need to select the switch type and whether ISDN accepts incoming voice or data calls. If you configure a PRI, you will need to configure the T1 or E1 controller. See the chapter "Configuring ISDN BRI" in the "ISDN Configuration" part of this guide, and the chapter "Configuring ISDN PRI" in the "Signaling Configuration" part of this guide.
  - Configuring CAS is described in the chapter "Configuring ISDN PRI" in the Signaling Configuration part of this guide.

If you want to configure SS7, refer to Appendix G, "Configuring the Cisco SS7/C7 Dial Access Solution System," in the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

Comparison of NextPort SPE and MICA Modem Commands

Table 7 through Table 10 compare the MICA and SPE commands.

Table 7 EXEC Commands: NextPort to MICA Command Comparison

NextPort SPE Commands	Purpose	MICA Modem Commands
clear port	Clears specified ports.	**clear modem**
clear port log	Clears all log entries for specified ports.	**clear modem log**

Table 7 EXEC Commands: NextPort to MICA Command Comparison (continued)

NextPort SPE Commands	Purpose	MICA Modem Commands
clear spe	Reboots all specified SPEs. All calls will be torn down.	none
clear spe counters	Clears all statistics.	**clear modem counters**
clear spe log	Clears all log entries for specified SPEs.	**clear modem log**
show port config	Displays configuration parameters for the current active session.	**show modem config**
show port modem calltracker	Displays port-level information for an active modem.	**show modem calltracker**
show port modem log	Displays the events generated by the modem sessions.	**show modem log**
show port modem test	Displays port modem test results.	**show modem test**
show port operational-status	Displays statistics for the current active session.	**show modem operational-status**
show spe	Displays the SPE status.	—
show spe log	Displays the SPE system log.	—
show spe modem active	Displays the statistics of all active calls on specified SPEs.	**show modem**
show spe modem csr	Displays the call success rate (CSR) for the specified SPE.	**show modem**
show spe modem disconnect-reason	Displays all modem disconnect reasons for the specified SPEs.	**show modem call-stats**
show spe modem high speed	Displays the total number of connections negotiated within each modulation or coder-decoder (codec) for a specific range of SPEs.	**show modem speed**
show spe modem high standard	Displays the total number of connections negotiated within each high modulation or codec for a specific range of SPEs or for all the SPEs.	—
show spe modem low speed	Displays the connect-speeds negotiated within each low-speed modulation or codec for a specific range of SPEs or for all the SPEs.	**show modem speed**
show spe modem low standard	Displays the total number of connections negotiated within each low modulation or codec for a specific range of SPEs or for all the SPEs.	—
show spe modem summary	Displays the modem service history statistics for specific SPEs.	**show modem**
show spe version	Displays all MICA and NextPort firmware versions stored in flash memory and the firmware assigned to each SPE.	**show modem mapping**

Table 8 SPE Configuration Commands: NextPort to MICA Command Comparison

NextPort SPE Commands	Purpose	MICA Modem Commands
busyout	Busies out active calls.	**modem busyout**
firmware location *filename*	Specifies the firmware file to be upgraded.	Already implemented on the Cisco AS5300 and Cisco AS5800 platforms.
firmware upgrade	Specifies the upgrade method.	Already implemented on the Cisco AS5300 platform.
port modem autotest[1]	Enables modem autotest.	**modem autotest**
shutdown	Tears down all active calls on the specified SPEs.	**modem shutdown**
spe	Configures the SPE.	Already implemented on the Cisco AS5300 and Cisco AS5800 platforms.
spe call-record	Generates a modem call record at the end of each call.	**modem call-record**
spe country	Sets the system country code.	**modem country**
spe log-size	Sets the maximum log entries for each port.	**modem buffer-size**
spe poll	Sets the statistic polling interval.	**modem poll**

1. Cisco does not recommend the use of the **modem autotest** or **port modem autotest** command. These commands may produce unexpected results including modems being marked out of service and unscheduled reloads. These commands have been removed in Cisco IOS Release 12.3.

Table 9 Port Configuration Commands: NextPort to MICA Command Comparison

NextPort SPE Commands	Purpose	MICA Modem Commands
busyout	Busies out a port.	**modem busyout**
default	Compares the value of the command to its default value.	**default modem**
port	Configures the port range.	**modem range**
shutdown	Shuts down a port.	**modem shutdown**

Table 10 Global Configuration Commands: NextPort to MICA Command Comparison

NextPort SPE CLI Commands	Purpose	MICA Modem CLI Commands
ds0 busyout-threshold	Defines a threshold to maintain a balance between the number of digital signal level 0s (DS0s) and modems.	**modem busyout-threshold**

Configuring Cisco Integrated Modems Using Modem Attention Commands

This section provides information about using modem attention (AT) command sets to modify modem configuration. It contains the following sections:

- Using Modem Dial Modifiers on Cisco MICA Modems (As required)
- Changing Configurations Manually in Integrated Microcom Modems (As required)
- Configuring Leased-Line Support for Analog Modems (As required)

Using Modem Dial Modifiers on Cisco MICA Modems

Dial modifiers permit multistage dialing for outbound modem calling through public and private switched telephone networks (PSTNs).

Note For additional information about dial modifiers for the MICA modems, search Cisco.com for the publication *AT Command Set and Register Summary for MICA Six-Port Modules*.

The Cisco NAS Modem Health feature is enabled by arguments to the **ATD AT** command. The **AT** prefix informs the network access server modem that commands are being sent to it, and the **D** (dial string or dial) suffix dials a telephone number, establishing a connection. With NAS Modem Health feature, you can enter the dial modifiers listed in Table 11 after the **D** in your dial string: **X**, **W**, and the comma (**,**) character. These modifiers had been previously accepted without error but ignored in Cisco MICA modems on Cisco AS5300 and Cisco AS5800 universal access servers.

Table 11 Dial Modifiers for Cisco MICA Modems

Dial Modifier	Definition
X	Switches to in-band dual tone multifrequency (DTMF) mode for any subsequent digits remaining in the **ATD** string. The **X** dial modifier has been added to serve as a delimiter for the host when the dial string is processed. It allows Cisco MICA portware to be used in many environments that do not support DTMF dialing (for example, PRI).
W	Waits for dial tone and then switches to in-band DTMF mode for any subsequent digits remaining in the **ATD** string. The **W** dial modifier also acts as a delimiter between the primary and secondary sections of the dial string, so that no additional **X** modifier is needed. Once either an **X** or a **W** has been parsed in the dial string, any additional **X** modifiers are ignored. Additional **W** modifiers cause Cisco MICA modems to wait for a dial tone.
,	Delay: Number of seconds in S8. Default is 2 seconds. The comma (**,**) dial modifier is treated as a silent DTMF tone for the duration of seconds specified in S8. The comma is acted on only after the call switching module (CSM) has made the transition to DTMF mode, which requires that it either follow an **X** or a **W** in the dial string, or that the T1/E1 be configured for DTMF signaling.

In the following example dial string, the portion of the string before the **X** is dialed for the given line type used in your configuration. All digits after the **X** generate the appropriate DTMF tones.

```
atdT5550101x,,567
```

Changing Configurations Manually in Integrated Microcom Modems

You can change the running configuration of an integrated modem by sending individual modem AT commands. Manageable Microcom modems have an out-of-band feature, which is used to poll modem statistics and send AT commands. The Cisco IOS software uses a direct connect session to transfer information through this out-of-band feature. To send AT commands to a Microcom modem, you must permit a direct connect session for a specified modem, open a direct connect session, send AT commands to a modem, and clear the directly connected session from the modem when you are finished.

Open a direct connect session by entering the **modem at-mode** *slot*/*port* command in privileged EXEC mode. From here, you can send AT commands directly from your terminal session window to the internal Microcom modems. Most incoming or outgoing calls on the modems are not interrupted when you open a direct connect session and send AT commands. However, some AT commands interrupt a call—for example, the **ATH** command, which hangs up a call. Open and close one direct connect session at a time. Note that multiple open sessions slow down modem performance.

Refer to the AT command set that came with your router for a complete list of AT commands that you can send to the modems.

For Microcom modems, you can clear or terminate an active directly connected session in two ways:

- Press **Ctrl-C** after sending all AT commands as instructed by the system when you enter AT command mode.
- Enter a second Telnet session and execute the **clear modem at-mode** *slot*/*port* EXEC command. This method is used for closing a directly connected session that may have been mistakenly left open by the first Telnet session.

The following example illustrates use of the modem commands.

AT Mode Example for Integrated Modems

To establish a direct connect session to an internal or integrated modem (existing inside the router), such as the connection required for Microcom modems in the Cisco AS5200 access server, open a directly connected session with the **modem at-mode** command and then send an AT command to the specified modem. For example, the following example sends the AT command **at%v** to modem 1/1:

```
AS5200# modem at-mode 1/1
You are now entering AT command mode on modem (slot 1 / port 1).
Please type CTRL-C to exit AT command mode.
at%v

MNP Class 10 V.34/V.FC Modem Rev 1.0/85

OK
at\s

IDLE           000:00:00
LAST DIAL

NET ADDR:      FFFFFFFFFFFF
MODEM HW: SA 2W United States
4 RTS 5 CTS 6 DSR - CD 20 DTR - RI
MODULATION     IDLE
MODEM BPS      28800  AT%G0
MODEM FLOW     OFF    AT\G0
MODEM MODE     AUT    AT\N3
V.23 OPR.      OFF    AT%F0
AUTO ANS.      ON     ATS0=1
SERIAL BPS     115200 AT%U0
BPS ADJUST     OFF    AT\J0
```

```
SPT BPS ADJ.    0       AT\W0
ANSWER MESSGS   ON      ATQ0
SERIAL FLOW     BHW     AT\Q3
PASS XON/XOFF   OFF     AT\X0
PARITY          8N      AT
```

The modem responds with "OK" when the AT command you send is received.

Configuring Leased-Line Support for Analog Modems

Analog modems on the NM-8AM and NM-16AM network modules in the Cisco 2600 and 3600 series routers provide two-wire leased-line support for enterprise customers who require point-to-point connections between locations and for enterprise customers with medium to high data transfer requirements without access to other technologies or with access to only low-grade phone lines.

This feature works only with leased lines that provide loop current. Each modem used must have an RJ-11 connection to the PSTN.

Several features enhance the analog modem software:

- 2-wire leased-line support.
- Modem speeds up to 33.6 kbps with support for all current analog modem protocols, compression, and error correction techniques.
- Power-on autoconnect and loopback testing.
- Support for the maximum number of leased-line users without data transmission loss at distances up to 2 to 5 km.
- In-band and out-of-band monitoring.
- Support on all Cisco 2600 and Cisco 3600 series platforms and upgradability using Cisco IOS software.
- Compatibility with other major leased-line modem vendors.

To configure this support, configure one modem AT command (**AT&L**) and two AT registers with the **modemcap entry** command for the appropriate leased lines.

For leased line configuration using the **AT&L**{**0** | **1** | **2**}command:

- **0**—Disables the leased line (enables switched line; default).
- **1**—Enables the leased line. The modem initiates a leased line when dial and answer commands (**ATD** and **ATA**) are issued.
- **2**—Enables the leased line. The modem goes off hook automatically after T57 number of seconds in:
  - Originate mode if ATS0 is 0.
  - Answer mode if ATS0 is not equal to 0.

The following AT registers can also be set:

- **AT:T57**—Number of seconds before going off hook in leased-line mode when the command **AT&L2** is used (defaults to 6).
- **AT:T79**—Number of autoretrains before the modem is disconnected (defaults to 3).

For more information about using the AT command set with the modems on the NM-8AM and NM-16AM network modules in the Cisco 2600 and 3600 series routers, search Cisco.com for the publication *AT Command Set and Register Summary for Analog Modem Network Modules.*

To configure a modem for leased-line operation, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# modemcap entry modem-type-name:AA=S0=0&L2`	Sets the modemcap for leased-line operation for the originating modem.
Step 2	`Router(config)# modemcap entry modem-type-name:AA=S0=1&L2`	Sets the modemcap for leased-line operation for the answering modem.

The **show modemcap** command lists all the predefined modem types and any user-defined modemcaps that are currently configured on the router:

- If the leased line has been configured, the modemcap information will be available.
- If the leased line has not been configured, only the predefined modem types will be displayed.

The important setting for leased-line support is what is defined in the modemcap as the key configuration item and its application to the leased line. Consider the following command strings:

```
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
```

AA stands for autoanswer:

- The answering modem AA register is set to 1 (AA=S0=1) so that autoanswer is “on”.
- The originating modem AA register is set to 0 (AA=S0=0) so that autoanswer is “off”.

If the AA feature is used, both the originating and answering modem must be put into leased-line mode with the **&L2** AT command.

In the examples, the micro_LL_orig and micro_LL_ans strings are arbitrary text descriptions.

Note For the **modemcap entry** command, one of the predefined modem types may be used or a completely user-defined modemcap may be created. For leased line, no new modem type was added. Users may create their own modemcaps for leased-line functionality.

To configure the modem for leased-line operation, use the **modemcap entry** command. For each connection, each modem must be configured as an originator or answerer.

The following example shows modemcaps for a leased-line originator and answerer and their application to specific ports:

```
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
line 73
 no exec
 modem InOut
 modem autoconfigure type micro_LL_ans
 transport input all
line 74
 no exec
 modem InOut
 modem autoconfigure type micro_LL_orig
 transport input all
```

Note When Multilink PPP (MLP) is configured on a dialer interface, the dialer configuration has a default value of 2 minutes for dialer idle timeout. For leased-line connections, set the dialer idle timeout to infinity by adding **dialer idle-timeout 0** to the configuration.

Verifying the Analog Leased-Line Configuration

The following information is important for verifying or troubleshooting your configuration. The **show modem log** command displays the progress of leased-line connections. Here is an example log for a leased-line answerer. Note the "LL Answering" state and "LL Answer" in the "Direction" field of the connection report:

```
00:44:03.884 DTR set high
00:44:02.888 Modem enabled
00:43:57.732 Modem disabled
00:43:52.476 Modem State:LL Answering
00:43:52.476 CSM:event-MODEM_STARTING_CONNECT New
State-CSM_CONNECT_INITIATED_STATE
00:43:51.112 Modem State:Waiting for Carrier
00:43:43.308 Modem State:Connected
00:43:42.304 Connection:TX/RX Speed = 33600/33600,
Modulation = V34
Direction = LL Answer, Protocol = MNP, Compression =
V42bis
00:43:42.304 CSM:event-MODEM_CONNECTED New
State-CONNECTED_STATE
00:43:42.300 RS232:noCTS* DSR* DCD* noRI noRxBREAK
TxBREAK*
00:43:41.892 PPP mode active
00:43:41.892 Modem enabled
00:43:39.888 PPP escape maps set:TX map=00000000 RX
map=FFFFFFFF
00:43:39.724 PPP escape maps set:TX map=00000000 RX
map=000A0000
00:43:34.444 RS232:CTS* DSR DCD noRI noRxBREAK TxBREAK
00:43:11.716 Modem Analog Report:TX = -20, RX = -34,
Signal to noise = 61
```

Cisco 2600 and 3600 Series Analog Modem Leased-Line Support Examples

In the following examples, one Cisco 3620 router and one Cisco 3640 router are connected back-to-back using leased lines. The Cisco 3620 router has the originating configuration, and the Cisco 3640 router has the answering configuration.

In the dialer interface configuration, the **dialer idle-timeout 0** command is added to set the dialer idle timeout to be infinity. Otherwise the leased line will go down and up every 2 minutes because the default dialer interface idle timeout is 2 minutes.

Note Except for passwords and logins, the Cisco IOS command-line interface (CLI) is case-insensitive. For this document, an uppercase "L" has been used in the command examples to avoid confusion with the numeral "1".

Leased-Line Originating Configuration

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
```

```
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
!
interface Async33
  no ip address
encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer in-band
 dialer pool-member 1
 async default routing
 async dynamic routing
 async mode dedicated
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp direction callout
 ppp multilink
!
interface Dialer1
 ip address 10.1.24.1 255.255.255.0
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer remote-name sara40
 dialer pool 1
 dialer idle-timeout 0
 dialer max-call 4096
 no cdp enable
 ppp direction callout
 ppp multilink
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 transport input none
 line 33
 no exec
 modem InOut
 modem autoconfigure type micro_LL_orig
 transport input all
line aux 0
 exec-timeout 0 0
line vty 0 4
 exec-timeout 0 0
!
end
```

Leased-Line Answering Configuration

```
version 12.1
service timestamps debug uptime
service timestamps log uptime
!
modemcap entry micro_LL_orig:AA=S0=0&L2
modemcap entry micro_LL_ans:AA=S0=1&L2
!
interface Async73
 no ip address
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer in-band
```

```
 dialer pool-member 1
 async default routing
 async dynamic routing
 async mode dedicated
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp direction callout
 ppp multilink
!
interface Dialer1
 ip address 10.1.24.2 255.255.255.0
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 load-interval 30
 dialer remote-name sara20
 dialer pool 1
 dialer idle-timeout 0
 dialer load-threshold 1 either
 dialer max-call 4096
 no cdp enable
 ppp direction callout
 ppp multilink
!
dialer-list 1 protocol ip permit
line con 0
 exec-timeout 0 0
 transport input none
line 73
 no exec
 modem InOut
 modem autoconfigure type micro_LL_ans
 transport input all
line aux 0
 transport input all
 flowcontrol hardware
line vty 0 4
 exec-timeout 0 0
!
end
```

Configuring Modem Pooling

Modem pooling allows you to control which modem a call connects to, on the basis of dialed number identification service (DNIS). When modem pooling is not used, incoming and outgoing calls are arbitrarily assigned to modems. For example, consider a Cisco AS5300 access server loaded with a 4-port ISDN PRI card. After an analog modem call comes into the first PRI trunk, the call is greeted by a general pool of B channels and a general pool of modems. Any B channel can be connected to any modem in the access server. A random assignment takes place. Modem resources cannot be controlled.

Modem pooling assigns physical modems to a single DNIS. It enables you to create pools of physical modems in one access server, assign a unique DNIS to each modem pool, and set maximum simultaneous connect limits.

This feature is used for physically partitioning or virtually partitioning modems inside one network access server.

Modem pooling offers these benefits:

- A certain number of modem ports can be guaranteed per DNIS.
- Maximum simultaneous connection limits can be set for each DNIS.

The following restrictions apply:

- Modem pooling is not a solution for large-scale dial access. It cannot be used to create virtual modem pools across multiple access servers that are connected. Modem pooling is physically restricted to one access server.
- MICA and Microcom technology modems support modem pooling. However, only MICA modems support modem pooling for CT1 and CE1 configurations using CAS. To use modem pooling with CT1 or CE1 connections, you must reserve at least two modems in the default modem pool. These reserved modems decode DNIS before handing off calls to the modems assigned to modem pools.

  If you see many call failures appearing on the access server, try assigning more modems to the default pool. Use the **show modem** and **show modem summary** EXEC commands to display the modem call failure and success ratio.
- No MIBs support modem pooling.
- The same DNIS cannot exist in more than one modem pool.

Modem pooling is supported on the Cisco AS5300 access servers. To configure and manage modems, perform the tasks in the following sections; all tasks are optional and depend upon the needs of your system.

- Creating a Modem Pool (Required)
- Verifying Modem Pool Configuration (As required)

Creating a Modem Pool

You must first decide to physically partition or virtually partition your modems. For more information, see the previous section, "Configuring Modem Pooling." After you have made this decision, create a modem pool for a dial-in service or specific customer by using the following commands beginning in global configuration mode.

	Command	Purpose
Step 1	`Router(config)# modem-pool name`	Creates a modem pool and assigns it a name, and starts modem pool configuration mode.
Step 2	`Router(config-modem-pool)# pool-range number-number`	Assigns a range of modems to the pool. A hyphen (**-**) is required between the two numbers. The range of modems you can choose from is equivalent to the number of modems in your access server that are not currently associated with another modem pool.
Step 3	`Router(config-modem-pool)# called-number number [max-conn number]`	Assigns the DNIS to be used for this modem pool. The **max-conn** option specifies the maximum number of simultaneous connections allowed for this DNIS. If you do not specify a **max-conn** value, the default (total number of modems in the pool) is used.[1]
Step 4	`Router(config-modem-pool)# Ctrl-Z`	Returns to EXEC mode.

	Command	Purpose
Step 5	`Router# show configuration`	Displays the running configuration to verify the modem pool settings. Make changes accordingly.
Step 6	`Router# copy running-config startup-config`	Saves the running configuration to the startup configuration.

1. The DNIS string can have an integer x to indicate a "don't care" digit for that position, for example, 555010x.

Note If you have active modem calls on the access server before using modem pooling, modem pooling gracefully applies itself to the access server. Modem pooling first waits for active calls to hang up before assigning modems to modem pools and directing calls according to DNIS.

Verifying Modem Pool Configuration

To verify the modem configuration, enter the **show modem-pool** command to display the configuration. This command displays the structure and activity status for all the modem pools in the access server. See Table 12 for a description of each display field.

```
Router# show modem-pool

modem-pool: System-def-Mpool
modems in pool: 0   active conn: 0
0 no free modems in pool

modem-pool: v90service
modems in pool: 48  active conn: 46
 8 no free modems in pool
 called_party_number: 1234
   max conn allowed: 48, active conn: 46
   8 max-conn exceeded, 8 no free modems in pool

modem-pool: v34service
modems in pool: 48  active conn: 35
 0 no free modems in pool
 called_party_number: 5678
   max conn allowed: 48, active conn: 35
   0 max-conn exceeded, 0 no free modems in pool
```

Table 12 show modem-pool Field Descriptions

Field	Description
modem-pool	Name of the modem pool. In the previous example, there are three modem pools configured: System-def-Mpool, v90service, and v34service. To set the modem pool name, refer to the **modem-pool** command. All the modems not assigned to a modem pool are automatically assigned to the system default pool (displayed as System-def-Mpool).
modems in pool	Number of modems assigned to the modem pool. To assign modems to a pool, refer to the display and descriptions for the **pool-range** command.

Table 12 show modem-pool Field Descriptions (continued)

Field	Description
active conn	Number of simultaneous active connections for the specified modem pool or called party DNIS number.
no free modems in pool	Number of times incoming calls were rejected because there were no more free modems in the pool to accept the call.
called_party_number	Specified called party DNIS number. This is the number that the remote clients use to dial in to the access server. You can have more than one DNIS number per modem pool. To set the DNIS number, refer to the description for the **called-number** command.
max conn allowed	Maximum number of modems that a called party DNIS number can use, which is an overflow protection measure. To set this feature, refer to the description for the **called-number** command.
max-conn exceeded	Number of times an incoming call using this called party DNIS number was rejected because the **max-conn** *number* parameter specified by the **called-number** command was exceeded.

For modem pool configuration examples, see the section "Physical Partitioning with Dial-In and Dial-Out Scenario" later in this chapter.

Check the following if you are having trouble operating your modem:

- Make sure you have not configured the same DNIS for multiple pools.
- Make sure you have not placed the same modem in multiple pools.

Note Modem pools that use MICA or Microcom modems support incoming analog calls over ISDN PRI. However, only MICA modems support modem pooling for T1 and E1 configurations with CAS.

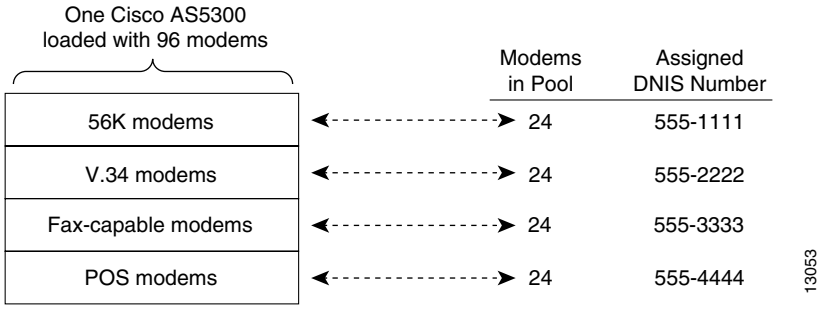
Configuring Physical Partitioning

You can either physically partition or virtually partition your modems to enable different dial-in and dial-out services. This section provides information about the following optional tasks:

- Creating a Physical Partition, page 86
- Physical Partitioning with Dial-In and Dial-Out Scenario, page 88

Physical partitioning uses one access server to function as multiple access servers loaded with different types of modem services (for example, V.34 modems, fax-capable modems, and point-of-sale (POS) modems). Each modem service is part of one physical modem pool and is assigned a unique DNIS number. (See Figure 19.)

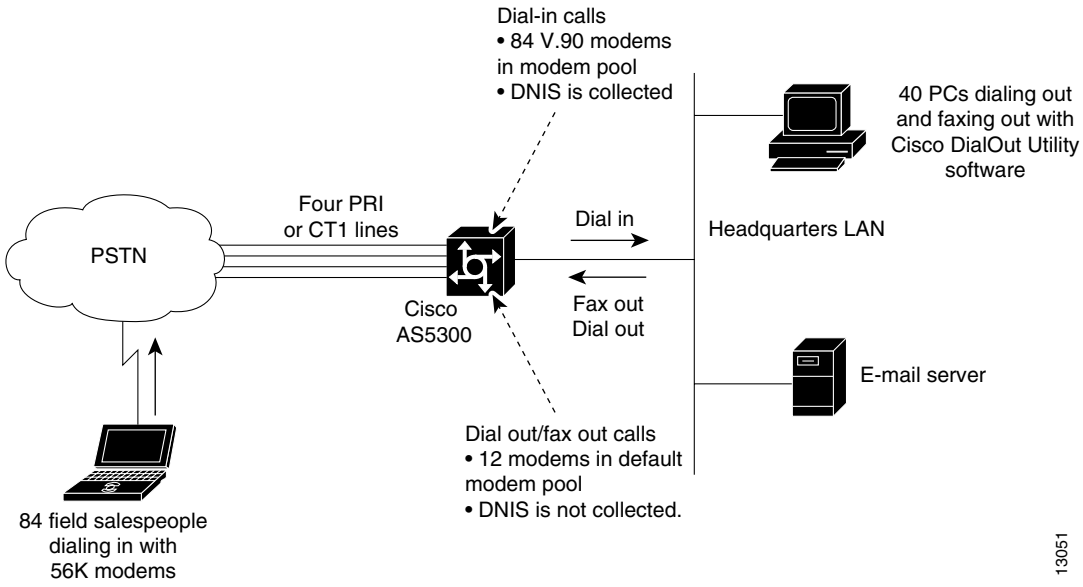
Figure 19 Modem Pooling Using Physical Partitioning



Physical partitioning can also be used to set up an access server for bidirectional dial access. (See Figure 20.)

Figure 20 shows one Cisco AS5300 access server loaded with 96 MICA modems and configured with 2 modem pools. One modem pool has 84 modems and collects DNIS. This pool is shared by 400 salespeople who remotely download e-mail from headquarters. The other modem pool contains 12 fax-capable modems and does not collect DNIS. This pool is shared by 40 employees using PCs on a LAN. Each time an outbound call is initiated by a PC, a modem on the Cisco AS5300 access server is seized and used to fax out or dial out. Not configuring DNIS support in the fax-out modem pool protects the pool from being used by the calls coming in from the field. Regardless of how many salespeople are dialing in or which telephone number they use, the fax-out and dial-out modem pool will always be reserved for the PCs connected to the LAN.

Figure 20 Modem Pooling Used for Bidirectional Dialing



Creating a Physical Partition

The following task creates one V.34 modem pool and one 56K modem pool on a Cisco AS5200. Each modem pool is configured with its own DNIS. Depending on which DNIS the remote clients dial, they connect to a 56K MICA modem or a V.34 Microcom modem.

The following hardware configuration is used on the Cisco AS5200 access server:

- One 2-port T1 PRI card
- One 48-port card containing four 6-port MICA 56K modem modules and two 12-port Microcom V.34 modem modules

To configure basic physical partitioning, perform the following steps:

Step 1 Enter global configuration mode:

```
Router# configure terminal
Router(config)#
```

Step 2 Create the modem pool for the 56K MICA modem services using the **modem-pool** *name* command. The modem pool is called 56kservices, which spans four 6-port MICA 56K modem modules.

```
Router(config)# modem-pool 56kservices
Router(config-modem-pool)#
```

Note The router is in modem pool configuration mode after the prompt changes from Router(config)# to Router(config-modem-pool)#.

Step 3 Assign a range of modems to the modem pool using the **pool-range** *number-number* command. Because all the 56K MICA technologies modems are seated in slot 1, they are assigned TTY line numbers 1 to 24. Use the **show line** EXEC command to determine the TTY line numbering scheme for your access server.

```
Router(config-modem-pool)# pool-range 1-24
```

Step 4 Assign a DNIS to the modem pool using the **called-number** *number* [**max-conn** *number*] command. This example uses the DNIS 5550101 to connect to the 56K modems. The maximum simultaneous connection limit is set to 24. The 25th user who dials 5550101 gets a busy signal.

```
Router(config-modem-pool)# called-number 5550101 max-conn 24
```

Step 5 Return to EXEC mode by entering **Ctrl-Z**. Next, display the modem pool configuration using the **show modem-pool** command. In the following example, 56K modems are in the modem pool called 56kservices. The remaining 24 V.34 Microcom modems are still in the default system pool.

```
Router(config-modem-pool)# ^Z
Router# show modem-pool

modem-pool: System-def-Mpool
modems in pool: 24   active conn: 0
0 no free modems in pool

modem-pool: 56kservices
modems in pool: 24  active conn: 0
 0 no free modems in pool
 called_party_number: 5550101
   max conn allowed: 24, active conn: 0
   0 max-conn exceeded, 0 no free modems in pool
```

Step 6 Create the modem pool for the Microcom physical partition. After the configuration is complete, the **show modem-pool** command shows that there are no remaining modems in the system default modem pool.

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# modem-pool v34services
```

```
Router(config-modem-pool)# pool-range 25-48
Router(config-modem-pool)# called-number 5550202 max-conn 24
Router(config-modem-pool)# ^Z
Router# show modem-pool

modem-pool: System-def-Mpool
modems in pool: 0   active conn: 0
0 no free modems in pool

modem-pool: 56kservices
modems in pool: 48  active conn: 0
 0 no free modems in pool
 called_party_number: 5550101
 max conn allowed: 48, active conn: 0
 0 max-conn exceeded, 0 no free modems in pool

modem-pool: v34services
modems in pool: 48  active conn: 0
 0 no free modems in pool
 called_party_number: 5550202
 max conn allowed: 48, active conn: 0
 0 max-conn exceeded, 0 no free modems in pool

Router# copy running-config startup-config
```

Physical Partitioning with Dial-In and Dial-Out Scenario

The following is a bidirectional dial scenario using a Cisco AS5300 access server. Two modem pools are configured. One modem pool contains 84 56K MICA modems, which is shared by 400 remote salespeople who dial in to headquarters. The other modem pool contains 12 fax-capable modems, which are shared by 40 employees who dial out of the headquarters LAN using the Cisco DialOut Utility software. See Figure 20 for the network topology.

The following hardware configuration is used on the Cisco AS5300:

- One 4-port T1 PRI card
- Two 48-port cards containing fourteen 6-port MICA 56K modem modules and two 6-port MICA fax-capable modem modules

To configure physical partitioning with dial-in and dial-out capability, perform the following steps:

Step 1 Create the 56K modem pool for the 400 remote salespeople. This modem pool contains 84 modems, which are reserved for the dial-in calls. To get access, the salespeople dial the DNIS 5550303. The total number of simultaneous calls is limited to 84. The 85th call and those above it are rejected. The **modem dialin** line configuration command is used to prevent modems 1 to 84 from dialing out.

```
Router# configure terminal
Router(config)# modem-pool 56ksalesfolks
Router(config-modem-pool)# pool-range 1-84
Router(config-modem-pool)# called-number 5550303 max-conn 84
Router(config-modem-pool)# exit
Router(config)# line 1 84
Router(config-line)# modem dialin
Router(config-line)# transport input all
Router(config-line)# rotary 1
Router(config-line)# autoselect ppp
Router(config-line)# exit
Router(config)#
```

Step 2 Create the dial-out/fax-out modem pool for the 40 local employees connected to the headquarters LAN. This modem pool contains 12 fax-capable MICA modems. No DNIS is assigned to the pool. Because lines 85 to 96 are used for the dial-out and fax-out modem services, the asynchronous lines are configured for reverse Telnet. This configuration is needed for the Telnet extensions to work with the dial-out application, which is installed on the LAN PCs.

```
Router(config)# modem-pool dialoutfolks
Router(config-modem-pool)# pool-range 85-96
Router(config-modem-pool)# exit
Router(config)# line 85-96
Router(config-line)# refuse-message z [!NMM!] No Modems Available z
Router(config-line)# exec-timeout 0 0
Router(config-line)# autoselect during-login
Router(config-line)# autoselect ppp
Router(config-line)# modem inout
Router(config-line)# rotary 1
Router(config-line)# transport preferred telnet
Router(config-line)# transport input all
Router(config-line)# exit
Router(config)#
```

Step 3 Configure the group asynchronous interface, which assigns core protocol characteristics to all the asynchronous interfaces in the system. Regardless of the direction that the modems are dialing, all modems in the access server leverage this group asynchronous configuration.

```
Router(config)# interface group-async 1
Router(config-if)# ip unnumbered ethernet 0
Router(config-if)# encapsulation ppp
Router(config-if)# async mode interactive
Router(config-if)# ppp authentication chap pap paplocal
Router(config-if)# peer default ip address pool bidir_dial_pool
Router(config-if)# no cdp enable
Router(config-if)# no ip mroute cache
Router(config-if)# no ip route cache
Router(config-if)# async dynamic routing
Router(config-if)# async dynamic address
Router(config-if)# group range 1-96
Building configuration...
Router(config-if)# exit
```

Step 4 Create an IP address pool for all the dial-in clients and dial-out clients. Both types of clients borrow addresses from this shared pool.

```
Router(config)# ip local pool bidir_dial_pool 10.4.1.1 10.4.1.96
Router(config)# ^z
Router# copy running-config startup-config
```

Step 5 (Optional) If you are using CiscoSecure AAA and a remote TACACS server, include the following security statements on the access server:

```
Router(config)# aaa new-model
Router(config)# aaa authentication login default tacacs+
Router(config)# aaa authentication login noaaa local
Router(config)# aaa authentication login logintac tacacs+
Router(config)# aaa authentication ppp ppptac tacacs+
Router(config)# aaa authentication ppp paplocal local
Router(config)# aaa authorization exec tacacs+
Router(config)# aaa authorization network tacacs+
Router(config)# aaa authorization reverse-access tacacs+
Router(config)# aaa accounting exec start-stop tacacs+
Router(config)# aaa accounting network start-stop tacacs+
Router(config)# aaa accounting update newinfo
Router(config)# enable password cisco
```

You should also include the host name, timeout interval, and authentication key:

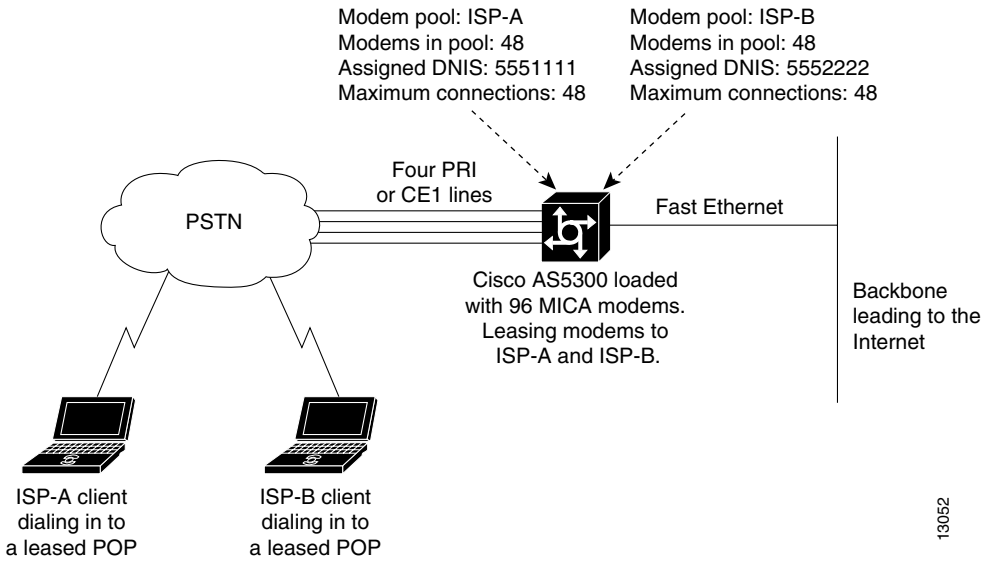
```
Router(config)# tacacs-server host 10.4.1.10
Router(config)# tacacs-server timeout 20
Router(config)# tacacs-server key nas1
```

Configuring Virtual Partitioning

Virtual partitioning creates one large modem pool on one access server, but assigns different DNIS numbers to different customers. Each incoming DNIS consumes resources from the same modem pool, but a maximum connect option is set for each DNIS.

Figure 21 shows two Internet service provider (ISP) customers who are leasing modems from another service provider. Each ISP is assigned its own DNIS number and range of modems. Each ISP is guaranteed a certain number of physical modem ports for simultaneous connections. After an ISP uses up all the modems assigned to its DNIS, a busy signal is issued.

Figure 21 Modem Pooling Using Virtual Partitioning



Virtual partitioning essentially resells modem banks to customers, such as a small-sized ISP. However, remember that modem pooling is a single-chassis solution, not a multichassis solution. Modem pooling is not a solution for reselling ports on a large-scale basis.

The following procedure creates one modem pool on a Cisco AS5300 access server for two ISP customers. The shared modem pool is called isp56kpool. However, both ISP customers are assigned different DNIS numbers and are limited to a maximum number of simultaneous connections.

See Figure 21 for the network topology.

The following hardware configuration is used on the Cisco AS5300 access server:

- One 4-port T1 PRI card
- Two 48-port cards containing sixteen 6-port MICA 56K modem modules

To configure virtual partitioning, perform the following steps:

Step 1 Enter global configuration mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

Step 2 Create the shared modem pool for the 56K MICA modem services. This modem pool is called isp56kpool, which spans sixteen 6-port MICA 56K modem modules.

```
Router(config)# modem-pool isp56kpool
Router(config-modem-pool)#
```

Step 3 Assign all the modems to the modem pool using the **pool-range** *number-number* command. Use the **show line** EXEC command to determine your TTY line numbering scheme.

```
Router(config-modem-pool)# pool-range 1-96
```

Step 4 Assign a unique DNIS to each ISP customer using the **called-number** *number* [**max-conn** *number*] command. In this example, the **max-conn** *number* option limits each ISP to 48 simultaneous connections. The 49th user to dial either DNIS will get a busy signal.

```
Router(config-modem-pool)# called-number 5550101 max-conn 48
Router(config-modem-pool)# called-number 5550202 max-conn 48
```

Step 5 Return to EXEC mode by entering a **Ctrl-Z** sequence. Next, display the modem pool configuration using the **show modem-pool** command. In the following example, all the 56K modems are in the isp56kpool modem pool. The output also shows two DNIS numbers configured: 5550101 and 5550202.

```
Router(config-modem-pool)# ^Z
Router# show modem-pool
modem-pool: System-def-Mpool
modems in pool: 0   active conn: 0
0 no free modems in pool

modem-pool: isp56kpool
modems in pool: 96  active conn: 0
 0 no free modems in pool
 called_party_number: 5550101
   max conn allowed: 48, active conn: 0
   0 max-conn exceeded, 0 no free modems in pool
 called_party_number: 5550202
   max conn allowed: 48, active conn: 0
   0 max-conn exceeded, 0 no free modems in pool
Router# copy running-config startup-config
```

Configuring Call Tracker

The Call Tracker feature captures detailed statistics on the status and progress of active calls and retains historical data for disconnected call sessions. Call Tracker collects session information such as call states and resources, traffic statistics, total bytes transmitted and received, user IP address, and disconnect reason. This data is maintained within the Call Tracker database tables, which are accessible through the Simple Network Management Protocol (SNMP), the CLI, or syslog.

Note The calltracker command, providing Call Tracker services, is supported for dial calls but not voice. Calltracker is supported for dial calls on 5x platforms (5300, 5350, 5400, 5800, and 5850).

Call Tracker is notified of applicable call events by related subsystems such as ISDN, PPP, CSM, Modem, EXEC, or TCP-Clear. SNMP traps are generated at the start of each call, when an entry is created in the active table, and at the end of each call, when an entry is created in the history table. Call Record syslogs are available through configuration that will generate detailed information records for all call terminations. This information can be sent to syslog servers for permanent storage and future analysis.

Additionally, the status and diagnostic data that is routinely collected from MICA modems is expanded to include new link statistics for active calls, such as the attempted transmit and receive rates, the maximum and minimum transmit and receive rates, and locally and remotely issued retrains and speedshift counters. For more detailed information on Call Tracker logs, refer to the TAC Tech Notes document, *Understanding Call Tracker Outputs*, at the following URL: http://www.cisco.com/warp/public/471/calltracker_view.html

To configure Call Tracker, perform the following steps:

	Command	Purpose
Step 1	`Router(config)# calltracker enable`	Enables Call Tracker.
Step 2	`Router(config)# calltracker call-record {terse|verbose} [quiet]`	Enables Call Tracker syslog support for generating detailed Call Records.
Step 3	`Router(config)# calltracker history max-size number`	Sets the maximum number of call entries to store in the Call Tracker history table.
Step 4	`Router(config)# calltracker history retain-mins minutes`	Sets the number of minutes for which calls are stored in the Call Tracker history table.
Step 5	`Router(config)# snmp-server packetsize byte-count`	Sets the maximum packet size allowed for SNMP server requests and replies.
Step 6	`Router(config)# snmp-server queue-length length`	Sets the queue length for SNMP traps.
Step 7	`Router(config)# snmp-server enable traps calltracker`	Enables Call Tracker to send traps whenever a call starts or ends.
Step 8	`Router(config)# snmp-server host host community-string calltracker`	Specifies the name or Internet address of the host to send Call Tracker traps.

Verifying Call Tracker

To verify the operation of Call Tracker, use the the following command in EXEC mode:

Command	Purpose
`Router# show call calltracker summary`	Verifies the Call Tracker configuration and current status.

Enabling Call Tracker

The following example shows how to enable the Call Tracker feature:

```
calltracker enable
```

```
calltracker call-record terse
calltracker history max-size 50
calltracker history retain-mins 5000
!
snmp-server engineID local 0012345
snmp-server community public RW
snmp-server community private RW
snmp-server community wxyz123 view v1default RO
snmp-server trap-source FastEthernet0
snmp-server packetsize 17940
snmp-server queue-length 200
snmp-server location SanJose
snmp-server contact Bob
snmp-server enable traps snmp
snmp-server enable traps calltracker
snmp-server enable traps isdn call-information
snmp-server enable traps hsrp
snmp-server enable traps config
snmp-server enable traps entity
snmp-server enable traps envmon
snmp-server enable traps bgp
snmp-server enable traps ipmulticast-heartbeat
snmp-server enable traps rsvp
snmp-server enable traps frame-relay
snmp-server enable traps rtr
snmp-server enable traps syslog
snmp-server enable traps dlsw
snmp-server enable traps dial
snmp-server enable traps dsp card-status
snmp-server enable traps voice poor-qov
snmp-server host 10.255.255.255 wxyz123
snmp-server host 10.0.0.0 xxxyyy calltracker
!
radius-server host 172.16.0.0 auth-port 1645 acct-port 1646 non-standard
radius-server key xyz
!
```

Configuring Polling of Link Statistics on MICA Modems

The status and diagnostic data that is routinely collected from MICA modems is expanded to include new link statistics for active calls, such as the attempted transmit and receive rates, the maximum and minimum transmit and receive rates, and locally and remotely issued retrains and speedshift counters. This connection data is polled from the modem at user-defined intervals and passed to Call Tracker.

To poll modem link statistics, use the following command in global configuration mode:

Command	Purpose
`Router(config)# modem link-info poll time` *seconds*	Sets the polling interval at which link statistics for active calls are retrieved from the modem.

Note The **modem link-info poll time** command consumes a substantial amount of memory, approximately 500 bytes for each MICA modem call. Use this command only if you require the specific data that it collects; for instance, if you have enabled Call Tracker on your access server.

Configuring MICA In-Band Framing Mode Control Messages

Dial-in Internet connections typically start in character mode to allow the user to log in and select a preferred service. When Cisco IOS software determines that the user wants a framed interface protocol during the call, such as PPP or SLIP, commands are sent to the MICA modem so that it will provide hardware assistance with the framing. This hardware assistance reduces the Cisco IOS processing load. To avoid loss or misinterpretation of framed data during the transition, issue these commands at precise times with respect to the data being sent and received.

MICA modem framing commands can be sent in the data stream itself, which greatly simplifies Cisco IOS tasks in achieving precision timing. For PPP connections, the common way for modems to connect to the Internet, total connect time might typically be improved by 2 to 3 seconds. This functionality reduces timeouts during PPP startup and reduces startup time. If an ASCII banner is sent just before PPP startup, this feature eliminates problems with banner corruption such as truncation and extraneous characters, thus improving the performance of terminal equipment.

In earlier software, the modem interface timing rules were not well understood and were difficult or impossible to implement using the separate command interface of the modem. The practical result is that the MICA in-band framing mode reduces the number of timeouts during PPP startup, and thus reduces startup time. MICA in-band framing is supported on MICA modems in Cisco AS5300 and Cisco AS5800 access servers.

To configure the MICA in-band framing mode control messages, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line line-number [ending-line-number]`	Specifies the number of modem lines to configure and enters line configuration mode. If a range is entered, it must be equal to the number of modems in the router.
Step 2	`Router(config-line)# no flush-at-activation`	Improves PPP and SLIP startup. Normally a router avoids line and modem noise by clearing the initial data received within the first one or two seconds. However, when the autoselect PPP feature is configured, the router flushes characters initially received and then waits for more traffic. This flush causes timeout problems with applications that send only one carriage return.

The Cisco IOS software offers additional interface commands that can be set to control modem interface timing. Refer to the Cisco IOS command references for more information about the interface commands described in the following paragraphs.

When a link goes down and comes back up before the timer set by the **carrier-delay** command expires, the down state is effectively filtered, and the rest of the software on the switch is not aware that a link-down event occurred. Therefore, a large carrier delay timer results in fewer link-up and link-down events being detected. On the other hand, setting the carrier delay time to 0 means that every link-up and link-down event is detected.

When the link protocol goes down (because of loss of synchronization, for example), the interface hardware is reset and the data terminal ready (DTR) signal is held inactive for at least the specified interval. Setting the **pulse-time** command enable pulsing DTR signal intervals on serial interfaces, and is useful for handling encrypting or other similar devices that toggle the DTR signal to resynchronize.

Use the **modem dtr-delay** command to reduce the time that a DTR signal is held down after an asynchronous line clears and before the DTR signal is raised again to accept new calls. Incoming calls may be rejected in heavily loaded systems, even when modems are unused because the default DTR hold-down interval may be too long. The **modem dtr-delay** command is designed for lines used for an unframed asynchronous session such as Telnet. Lines used for a framed asynchronous session such as PPP should use the **pulse-time** interface command.

Enabling Modem Polling

The following example enables modem status polling through the out-of-band feature, which is associated to line 1:

```
Router# configure terminal
Router(config)# line 1
Router(config-line)# modem status-poll
```

Setting Modem Poll Intervals

The following example sets the time interval between polls to 10 seconds using the **modem poll time global** configuration command:

```
Router# configure terminal
Router(config)# modem poll time 10
```

Setting Modem Poll Retry

The following example configures the server to attempt to retrieve statistics from a local modem up to five times before discontinuing the polling effort:

```
Router# configure terminal
Router(config)# modem poll retry 5
```

Collecting Modem Statistics

Depending upon your modem type, the Cisco IOS software provides several **show** EXEC commands that allow you to display or poll various modem statistics. See Table 7 and Table 8 to find the **show** EXEC command appropriate for your modem type and the task you want to perform.

Logging EIA/TIA Events

To facilitate meaningful analysis of the modem log, turn the storage of specific types of EIA/TIA events on or off. To activate or inactivate the storage of a specific type of EIA/TIA modem event for a specific line or set of lines, use either of the following commands in line configuration mode, as needed:

Command	Purpose
`Router(config-line)# modem log {cts | dcd | dsr | dtr | ri | rs323 | rts | tst}`	Configures the types of EIA/TIA events that are stored in the modem log. The default setting stores no EIA/TIA events.
or	
`Router(config-line)# no modem log {cts | dcd | dsr | dtr | ri | rs323 | rts | tst}`	Turns off the logging of a specific type of EIA/TIA event.

Configuring a Microcom Modem to Poll for Statistics

Manageable Microcom modems have an out-of-band feature, which is used for polling modem statistics. To configure the system to poll for modem statistics, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# modem poll time` *seconds*	Specifies the number of seconds between statistical modem polling for Microcom modems. The default is 12 seconds. The configuration range is from 2 to 120 seconds.
Step 2	`Router(config)# modem poll retry` *number*	Sets the maximum number of polling attempts to Microcom modems. The default is three polling attempts. The configuration range is from 0 to 10 attempts.[1]
Step 3	`Router(config)# modem status-poll`	Polls for status and statistics for a Microcom modem through the modem's out-of-band feature.
Step 4	`Router(config)# modem buffer-size` *number*	Defines the number of modem events that each modem is able to store. The default is 100 events for each modem. Use the **show modem log** command to display modem events.

1. If the number of attempts to retrieve modem status or statistics exceeds the number you define, the out-of-band feature is removed from operation. In this case, you must reset the modem hardware using the **clear modem** command.

Troubleshooting Using a Back-to-Back Modem Test Procedure

You can manually isolate an internal back-to-back connection and data transfer between two modems for focused troubleshooting purposes. For example, if mobile users cannot dial in to modem 2/5 (which is the sixth modem port on the modem board in the second chassis slot), attempt a back-to-back test with modem 2/5 and a modem known to be functioning, such as modem 2/6. You might need to enable this command on several different combinations of modems to determine which one is not functioning properly. A pair of operable modems connect and complete sending data in both directions. An operable modem and an inoperable modem do not connect with each other.

To perform the modem test procedure, enter the **test modem back-to-back** *first-slot/port second-slot/port* command, as follows:

Step 1 Perform a back-to-back modem test between two normal functioning modems. This example shows a successful connection between modem 1/1 and modem 1/0, which verifies normal operating conditions between these two modems:

```
Router# test modem back-to-back 1/1 1/0
Repetitions (of 10-byte packets) [1]: 10
Router#
%MODEM-5-B2BCONNECT: Modems (1/1) and (1/0) connected in back-to-back test: CONN
ECT9600/REL-MNP
%MODEM-5-B2BMODEMS: Modems (1/0) and (1/1) completed back-to-back test: success/
packets = 20/20
```

After you enter the **test modem back-to-back** command, you must define the number of packets sent between modems at the Repetitions prompt. The ideal range of packets to send and receive is from 1 to 100. The default is 1 packet that is 10 bytes large. The response message (for example, "success/packets = 20/20") tells you how many packets were sent in *both* directions compared to the total number of packets attempted to be sent in both directions. Because the software reports the packet total in both directions, the reported numbers are *two times* the number you originally specify.

When a known good modem is tested against a known bad modem, the back-to-back modem test fails. In the following example, modem 1/3 is suspected or proven to be inoperable or bad:

```
Router# test modem back-to-back 1/1 1/3
Repetitions (of 10-byte packets) [1]: 10
Router#
%MODEM-5-BADMODEMS: Modems (1/3) and (1/1) failed back-to-back test: NOCARRIER
```

Step 2 You would need to manually mark modem 1/3 as an inoperable or bad modem. You mark the bad modem by determining which line number corresponds with the modem. Use the **show modem 1/3** EXEC command to verify that TTY line number 4 (shown as `TTY4`) is used for modem 1/3:

```
Router# show modem 1/3
Mdm  Typ    Status     Tx/Rx     G  Duration  TX  RX  RTS  CTS  DSR  DCD  DTR
1/3  V34    Idle    28800/28800  0  00:00:00           x    x    x    x    x

Modem 1/3, Microcom MNP10 V34 Modem (Managed), TTY4
Firmware (Boot) Rev: 1.0(23) (1.0(5))
Modem config: Incoming and Outgoing
Protocol: reliable/MNP, Compression: V42bis
Management port config: Status polling and AT session
Management port status: Status polling and AT session
TX signals: -15 dBm, RX signals: -17 dBm


  Last clearing of "show modem" counters never
    0 incoming completes, 0 incoming failures
    0 outgoing completes, 0 outgoing failures
    0 failed dial attempts, 0 ring no answers, 1 busied outs
    0 no dial tones, 0 dial timeouts, 0 watchdog timeouts
    0 no carriers, 0 link failures, 0 resets, 0 recover oob
    0 protocol timeouts, 0 protocol errors, 0 lost events

Transmit Speed Counters:

Connection Speeds         75      300      600     1200     2400     4800
# of connections           0        0        0        0        0        0
Connection Speeds       7200     9600    12000    14400    16800    19200
# of connections           0        0        0        0        0        0
Connection Speeds      21600    24000    26400    28800    31200    32000
# of connections           0        0        0        1        0        0
Connection Speeds      33600    34000    36000    38000    40000    42000
# of connections           0        0        0        0        0        0
Connection Speeds      44000    46000    48000    50000    52000    54000
# of connections           0        0        0        0        0        0
Connection Speeds      56000
# of connections           0
```

Step 3 Enter line configuration mode and manually remove modem 1/3 from dial services by entering the **modem bad** command on line 4:

```
Router# configure terminal
Router(config)# line 4
Router(config-line)# modem bad
Router(config-line)# exit
Router(config)# exit
```

Step 4 Enter the **show modem** EXEC command or the **show modem** *slot/port* command to display the bad modem status.

Bad modems are marked with the letter B in the Mdm column of the **show modem** command display output.

```
Router# show modem

%SYS-5-CONFIG_I: Configured from console by consolem
                 Inc calls      Out calls   Busied   Failed  No       Succ
  Mdm   Usage    Succ   Fail    Succ   Fail   Out      Dial    Answer   Pct.
  1/0     0%       0      0       0      0       1        0       0      0%
  1/1     0%       0      0       0      0       3        0       0      0%
  1/2     0%       0      0       0      0       1        0       0      0%
B 1/3     0%       0      0       0      0       1        0       0      0%
  1/4     0%       0      0       0      0       1        0       0      0%
  1/5     0%       0      0       0      0       1        0       0      0%
  1/6     0%       0      0       0      0       1        0       0      0%
  1/7     0%       0      0       0      0       1        0       0      0%
  1/8     0%       0      0       0      0       1        0       0      0%
  1/9     0%       0      0       0      0       1        0       0      0%
  1/10    0%       0      0       0      0       1        0       0      0%
  1/11    0%       0      0       0      0       1        0       0      0%
  1/12    0%       0      0       0      0       1        0       0      0%
  1/13    0%       0      0       0      0       1        0       0      0%
  1/14    0%       0      0       0      0       1        0       0      0%
  1/15    0%       0      0       0      0       1        0       0      0%
  1/16    0%       0      0       0      0       1        0       0      0%
  1/17    0%       0      0       0      0       1        0       0      0%
  1/18    0%       0      0       0      0       0        0       0      0%
  1/19    0%       0      0       0      0       0        0       0      0%
  1/20    0%       0      0       0      0       0        0       0      0%
  1/21    0%       0      0       0      0       0        0       0      0%
  1/22    0%       0      0       0      0       0        0       0      0%
  1/23    0%       0      0       0      0       0        0       0      0%
```

Malfunctioning modems are also marked as Bad in the Status column of the **show modem** *slot/port* command display output, as the following example shows:

```
Router# show modem 1/3

Mdm  Typ    Status     Tx/Rx     G  Duration  TX  RX  RTS  CTS  DSR  DCD  DTR
1/3  V34    Bad     28800/28800  0  00:00:00           x    x    x    x    x

Modem 1/3, Microcom MNP10 V34 Modem (Managed), TTY4
Firmware (Boot) Rev: 1.0(23) (1.0(5))
Modem config: Incoming and Outgoing
Protocol: reliable/MNP, Compression: V42bis
Management port config: Status polling and AT session
Management port status: Status polling and AT session
TX signals: -15 dBm, RX signals: -17 dBm

  Last clearing of "show modem" counters never
    0 incoming completes, 0 incoming failures
    0 outgoing completes, 0 outgoing failures
```

```
    0 failed dial attempts, 0 ring no answers, 1 busied outs
    0 no dial tones, 0 dial timeouts, 0 watchdog timeouts
    0 no carriers, 0 link failures, 0 resets, 0 recover oob
    0 protocol timeouts, 0 protocol errors, 0 lost events

Transmit Speed Counters:

Connection Speeds          75      300      600     1200     2400     4800
# of connections            0        0        0        0        0        0
Connection Speeds        7200     9600    12000    14400    16800    19200
# of connections            0        0        0        0        0        0
Connection Speeds       21600    24000    26400    28800    31200    32000
# of connections            0        0        0        1        0        0
Connection Speeds       33600    34000    36000    38000    40000    42000
# of connections            0        0        0        0        0        0
Connection Speeds       44000    46000    48000    50000    52000    54000
# of connections            0        0        0        0        0        0
Connection Speeds       56000
# of connections            0
```

Clearing a Direct Connect Session on a Microcom Modem

The examples in this section are for Microcom modems.

The following example shows how to execute the **modem at-mode** command from a Telnet session:

```
Router# modem at-mode 1/1
```

The following example shows how to execute the **clear modem at-mode** command from a second Telnet session while the first Telnet session is connected to the modem:

```
Router# clear modem at-mode 1/1
clear "modem at-mode" for modem 1/1 [confirm] <press Return>
Router#
```

The following output is displayed in the first Telnet session after the modem is cleared by the second Telnet session:

```
Direct connect session cleared by vty0 (172.19.1.164)
```

Displaying Local Disconnect Reasons

To find out why a modem ended its connection or why a modem is not operating at peak performance, use the **show modem call-stats** [*slot*] EXEC command.

Disconnect reasons are described using four hexadecimal digits. The three lower-order digits can be used to identify the disconnect reason. The high-order digit generally indicates the type of disconnect reason or the time at which the disconnect occurred. For detailed information on the meaning of hexadecimal values for MICA modem disconnects, refer to the TAC Tech Notes document, *MICA Modem States and Disconnect Reasons*, at the following URL: http://www.cisco.com/warp/public/76/mica-states-drs.html

For detailed information on the meaning of hexadecimal values for NextPort modem disconnects, refer to the TAC Tech Notes document, *Interpreting NextPort Disconnect Reason Codes*, at the following URL: http://www.cisco.com/warp/public/471/np_disc_code.html .

Local disconnect reasons are listed across the top of the screen display (for example, wdogTimr, compress, retrain, inacTout, linkFail, moduFail, mnpProto, and lapmProt). In the body of the screen display, the number of times each modem disconnected is displayed (see the # column). For a particular disconnect reason, the % column indicates the percent that a modem was logged for the specified disconnect reason with respect to the entire modem pool for that given reason. For example, out of all the times the rmtLink error occurred on all the modems in the system, the rmtLink error occurred 10 percent of the time on modem 0/22.

Malfunctioning modems are detected by an unusually high number of disconnect counters for a particular disconnect reason. For example, if modem 1/0 had a high number of compression errors compared to the remaining modems in system, modem 1/0 would likely be the inoperable modem.

To reset the counters displayed by the **show modem call-stats** command, enter the **clear modem counters** command.

Note For a complete description of each error field displayed by the commands on this page, refer to the *Cisco IOS Dial Technologies Command Reference*. Remote disconnect reasons are not described by the **show modem** command output.

The following example displays output for the **show modem call-stats** command. Because of the screen size limitation of most terminal screen displays, not all possible disconnect reasons are displayed at one time. Only the top eight most frequently experienced disconnect reasons are displayed at one time.

```
Router# show modem call-stats

  dial-in/dial-out call statistics

       lostCarr  dtrDrop  rmtLink wdogTimr compress  retrain inacTout linkFail
  Mdm     #   %    #   %    #   %    #   %    #   %    #   %    #   %    #   %
* 0/0     6   2    2   3    1   0    0   0    0   0    0   0    0   0    0   0
* 0/1     5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
  0/2     5   2    2   3    4   3    0   0    0   0    0   0    0   0    0   0
* 0/3     5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/4     5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/5     5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/6     4   1    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/7     4   1    2   3    4   3    0   0    0   0    0   0    0   0    0   0
* 0/8     6   2    1   1    3   2    0   0    0   0    0   0    0   0    0   0
* 0/9     5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/10    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 0/11    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
  0/12    5   2    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 0/13    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/14    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/15    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/16    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/17    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 0/18    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 0/19    5   2    1   1    3   2    0   0    0   0    0   0    0   0    0   0
* 0/20    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/21    5   2    1   1    1   0    0   0    0   0    0   0    0   0    0   0
* 0/22    5   2    1   1   11  10    0   0    0   0    0   0    0   0    0   0
* 0/23    5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/0     4   1    2   3    2   1    0   0    0   0    0   0    0   0    0   0
* 2/1     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/2     5   2    2   3    0   0    0   0    0   0    0   0    0   0    0   0
* 2/3     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/4     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/5     5   2    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/6     4   1    1   1    1   0    0   0    0   0    0   0    0   0    0   0
```

```
* 2/7      5   2     1   1     1   0     0   0     0   0     0   0     0   0     0   0
* 2/8      5   2     1   1     1   0     0   0     0   0     0   0     0   0     0   0
* 2/9      4   1     1   1     2   1     0   0     0   0     0   0     0   0     0   0
* 2/10     5   2     1   1     0   0     0   0     0   0     0   0     0   0     0   0
* 2/11     5   2     1   1     5   4     0   0     0   0     0   0     0   0     0   0
* 2/12     5   2     1   1     2   1     0   0     0   0     0   0     0   0     0   0
* 2/13     5   2     1   1     1   0     0   0     0   0     0   0     0   0     0   0
* 2/14     5   2     1   1     2   1     0   0     0   0     0   0     0   0     0   0
* 2/15     4   1     1   1     3   2     0   0     0   0     0   0     0   0     0   0
* 2/16     4   1     1   1     3   2     0   0     0   0     0   0     0   0     0   0
* 2/17     5   2     2   3     9   8     0   0     0   0     0   0     0   0     0   0
* 2/18     4   1     1   1     1   0     0   0     0   0     0   0     0   0     0   0
* 2/19     3   1     1   1     2   1     0   0     0   0     0   0     0   0     0   0
* 2/20     7   3     1   1     8   7     0   0     0   0     0   0     0   0     0   0
* 2/21     5   2     1   1     1   0     0   0     0   0     0   0     0   0     0   0
* 2/22     4   1     1   1     2   1     0   0     0   0     0   0     0   0     0   0
* 2/23     5   2     1   1     2   1     0   0     0   0     0   0     0   0     0   0
 Total   233       59       110       0       0       0       0       0       0
```

dial-out call statistics

```
          noCarr noDitone      busy     abort dialStrg autoLgon dialTout  rmtHgup
  Mdm      #   %    #   %    #   %    #   %    #   %    #   %    #   %    #   %
* 0/0      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/1      0   0    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   0
* 0/3      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/4      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/5      0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/6      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/7      5   5    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/8      0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/9      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/10     0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/11     5   5    0   0    0   0    0   0    0   0    0   0    0   0    0   0
  0/12     0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/13     0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/14     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/15     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/16     2   2    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/17     4   4    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/18     5   5    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/19     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/20     0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/21     0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/22     5   5    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 0/23     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/0      2   2    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/1      3   3    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/2      0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/3      0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/4      0   0    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/5      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/6      1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/7      4   4    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/8      7   8    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/9      4   1    1   1    2   1    0   0    0   0    0   0    0   0    0   0
* 2/10     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/11     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/12     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/13     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/14     4   4    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/15     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
* 2/16     1   1    0   0    0   0    0   0    0   0    0   0    0   0    0   0
```

```
* 2/17     5    5     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/18     5    5     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/19     3    3     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/20     0    0     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/21     4    4     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/22     2    2     0    0     0    0     0    0     0    0     0    0     0    0     0    0
* 2/23     0    0     0    0     0    0     0    0     0    0     0    0     0    0     0    0
  Total    84         0         0         0         0         0         0         0         0
```

Removing Inoperable Modems

To manually remove inoperable modems from dialup services, use the following commands in line configuration mode:

	Command	Purpose
Step 1	`Router(config-line)# modem bad`	Removes and idles the modem from service and indicates it as suspected or proven to be inoperable.
Step 2	`Router(config-line)# modem hold-reset`	Resets and isolates the modem hardware for extensive troubleshooting.
Step 3	`Router(config-line)# modem shutdown`	Abruptly shuts down a modem from dial service.
Step 4	`Router(config-line)# modem recovery-time` *minutes*	Sets the maximum amount of time for which the call-switching module waits for a local modem to respond to a request before it is considered locked in a suspended state. The default is 5 minutes.

If you use the **modem bad** command to remove an idle modem from dial services and mark it as inoperable, the letter B is used to identify the modem as bad. The letter B appears in the Status column in the output of **show modem** *slot*/*port* command and in the far left column in the output of the **show modem** command. Use the **no modem bad** command to unmark a modem as B and restore it for dialup connection services. If the letter B appears next to a modem number, it means the modem was removed from service with the **modem shutdown** command.

Note Only idle modems can be marked "bad" by the **modem bad** command. If you want to mark a modem bad that is actively supporting a call, first enter the **modem shutdown** command, then enter the **modem bad** command.

Use the **modem hold-reset** command if a router is experiencing extreme modem behavior (for example, if the modem is uncontrollably dialing in to the network). This command prevents the modem from establishing software relationships such as those created by the **test modem back-to-back** command. The modem is unusable while the **modem hold-reset** command is configured. The **modem hold-reset** command also resets a modem that is frozen in a suspended state. Disable the suspended modem with the **modem hold-reset** command, and then restart hardware initialization with the **no modem hold-reset** command.

The following example disables a suspended modem and resets its hardware initialization:

```
Router# configure terminal
Router(config)# line 4
Router(config-line)# modem hold-reset
Router(config-line)# no modem hold-reset
```

The following example gracefully disables the modem associated with line 1 from dialing and answering calls. The modem is disabled only after all active calls on the modem are dropped.

```
Router# configure terminal
Router(config)# line 1
Router(config)# modem busyout
```

The following example abruptly shuts down the modem associated with line 2. All active calls on the modem are dropped immediately.

```
Router# configure terminal
Router(config)# line 2
Router(config)# modem shutdown
```

In the following example, the modem using TTY line 3 is actively supporting a call (as indicated by the asterisk). However, we want to mark the modem bad because it has poor connection performance. First, abruptly shut down the modem and drop the call with the **modem shutdown** command, and then enter the **modem bad** command to take the modem out of service.

```
Router# show modem

                Inc calls     Out calls     Busied   Failed  No       Succ
  Mdm  Usage    Succ   Fail   Succ   Fail   Out      Dial    Answer   Pct.
  1/0    37%     98      4      0      0       0        0       0     96%
  1/1    38%     98      2      0      0       0        0       0     98%
* 1/2    2%      3       99     0      0       0        0       0     1%
.
.
.

Router# configure terminal
Router(config)# line 3
Router(config)# modem shutdown
Router(config)# modem bad
Router(config)# exit

Router# show modem

                Inc calls     Out calls     Busied   Failed  No       Succ
  Mdm  Usage    Succ   Fail   Succ   Fail   Out      Dial    Answer   Pct.
  1/0    37%     98      4      0      0       0        0       0     96%
  1/1    38%     98      2      0      0       0        0       0     98%
B 1/2    2%      3       99     0      0       0        0       0     1%
```

For more information about modem recovery procedures, refer to TAC Tech Notes *Configuring MICA Modem Recovery* at http://www.cisco.com/warp/public/76/modem-recovery.html and *Configuring NextPort SPE Recovery* at http://www.cisco.com/warp/public/76/spe-recovery.html.

Busying Out a Modem Card

To busy out a modem card in a Cisco access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line shelf/slot/port`	Specifies the line number, by specifying the shelf, slot, and port numbers; you must type in the slashes. This command also begins line configuration mode.
Step 2	`Router(config-line)# modem busyout`	Having specified the modem to be busied out with the **line** command, enter the **modem busyout** command to busy out the modem. The command disables the modem associated with line *shelf/slot/port* from dialing and answering calls.You need not specify a *shelf/slot/port* number again in this command.
Step 3	`Router(config-line)# modem shutdown`	Having specified the modem to be shut down with the **line** command, enter the **modem shutdown** command to shut down the modem, whether or not it has already been busied out. You need not specify a *shelf/slot/port* number again in this command because you have already done so with the **line** command.
Step 4	`Router(config-line)# exit`	Exits line configuration mode and returns to global configuration mode.
Step 5	`Router(config)# modem busyout-threshold number`	Specifies a threshold number using the **modem busyout-threshold** *number* command to balance the number of DS0s with the number of modem lines. For more information, refer to the *Cisco IOS Dial Technologies Command Reference*.
Step 6	`Router(config)# exit`	Exits global configuration mode and returns to privileged EXEC mode.
Step 7	`Router# show busyout`	From privileged EXEC mode, verifies that the line is busied out. If there are active calls, the software waits until the call terminates before the line is busied out.

The **modem busyout** command disables the modem associated with a specified line from dialing and answering calls. The **modem busyout** command can busy out and eventually terminate all 72 ports on the Cisco AS5800 modem card.

Monitoring Resources on Cisco High-End Access Servers

The following tasks enable you to monitor the network access server (NAS) health conditions at the DS0 level, PRI bearer channel level, and modem level. Performing these tasks will benefit network operation with improved visibility into the line status for the NAS for comprehensive health monitoring and notification capability, and improved troubleshooting and diagnostics for large-scale dial networks.

Perform the following tasks to monitor resource availability on the Cisco high-end access servers:

- Enabling DS0 Busyout Traps—DS0 busyout traps are generated when there is a request to busy out a DS0, when there is a request to take a DS0 out of busyout mode, or when busyout completes and the DS0 is out-of-service. DS0 busyout traps are generated at the DS0 level for both CAS and ISDN

configured lines. This feature is enabled and disabled through use of the CLI and MIBs. DS0 busyout traps are disabled by default and are supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 universal access servers.

- Enabling ISDN PRI Requested Channel Not Available Traps—ISDN PRI channel not available traps are generated when a requested DS0 channel is not available, or when there is no modem available to take the incoming call. This feature is available only for ISDN PRI interfaces. This feature is enabled and disabled through use of CLI for ISDN traps and the CISCO-ISDN-MIB. ISDN PRI channel not available traps are disabled by default and are supported on the Cisco AS5300, Cisco AS5400, and Cisco AS5800.
- Enabling 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. This feature is enabled and disabled through use of CLI and the CISCO-MODEM-MGMT-MIB. Modem health traps are disabled by default and are supported on the Cisco AS5300, Cisco AS5400, and Cisco AS5800.
- Enabling DS1 Loopback Traps—DS1 loopback traps are generated when a DS1 line goes into loopback mode. This feature is enabled and disabled by CLI and the CISCO-POP-MGMT-MIB. DS1 loopback traps are disabled by default and are supported on the Cisco AS5300 and Cisco AS5400 only.

The CISCO-POP-MGMT-MIB supplies the DS0 busyout traps and the DS1 loopback traps. The CISCO-MODEM-MGMT-MIB supplies additional modem health traps when the modem port becomes non-functional. The CISCO-ISDN-MIB supplies additional traps for ISDN PRI channel not available.

To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

See the sections "Verifying Enabled Traps" and "Troubleshooting the Traps" to verify and troubleshoot configuration. The section "NAS Health Monitoring Example" provides output of a configuration with the NAS health monitoring features enabled.

Enabling DS0 Busyout Traps

Before you enable DS0 busyout traps, the SNMP manager must already have been installed on your workstation, and the SNMP agent must be configured on the NAS by entering the **snmp-server community** and **snmp-server host** commands. Refer to the *Cisco IOS Configuration Fundamentals Configuration Guide* for more information on these commands.

To generate DS0 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 DS0 or to indicate when busyout finishes.

Enabling ISDN PRI Requested Channel Not Available Traps

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 NAS rejects an incoming call on an ISDN PRI interface because the channel is not available.

Enabling Modem Health Traps

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 prepared for firmware download; when download fails; when placed in loopback mode for maintenance; or when there is a request to busy out the modem.

Enabling DS1 Loopback Traps

To generate DS1 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 DS1 line goes into loopback mode.

Verifying Enabled Traps

To verify that the traps are enabled, use the **show run** command. The following output indicates that all the traps are enabled:

```
Router(config)# show run

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
```

Additionally, you can use the **show controllers** command with the **timeslots** keyword to display details about the channel state. This feature shows whether the DS0 channels of a particular controller are in idle, in-service, maintenance, or busyout state. This enhancement applies to both CAS and ISDN PRI interfaces and is supported on the Cisco AS5300 and Cisco AS5400 only.

Troubleshooting the Traps

To troubleshoot the traps, turn on the debug switch for SNMP packets by entering the following command in privileged EXEC mode:

```
Router# debug snmp packets
```

Check the resulting output to see that the SNMP trap information packet is being sent. The output will vary based on the kind of packet sent or received:

```
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 like snmptrapd, with debugging flags turned on, to monitor output.

NAS Health Monitoring Example

The following is sample configuration output showing all NAS health monitoring traps turned on:

```
Building configuration...

Current configuration:
! Last configuration change at 12:27:30 pacific Thu May 25 2000
version xx.x
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
!
controller T1 2
 shutdown
 clock source line secondary 2
!
controller T1 3
 shutdown
 clock source line secondary 3
!
controller T1 4
 shutdown
 clock source line secondary 4
!
controller T1 5
 shutdown
 clock source line secondary 5
!
controller T1 6
 shutdown
 clock source line secondary 6
!
controller T1 7
 shutdown
 clock source line secondary 7
!
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 Serial2
 no ip address
 shutdown
!
interface Serial3
 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 ds0-busyout
snmp-server enable traps isdn chan-not-avail
snmp-server enable traps modem-health
snmp-server enable traps ds1-loopback
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
```

Configuration Examples for Modem Management

This section provides the following examples:

- NextPort Modem Log Example
- Modem Performance Summary Example
- Modem AT-Mode Example
- Connection Speed Performance Verification Example

For additional information and examples about the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*.

NextPort Modem Log Example

The following is partial sample output for the Cisco AS5400 with the NextPort Distributed forwarding Card (DFC). This example shows the port history event log for slot 5, port 47:

```
Router# show port modem log 5/47

Port 5/47 Events Log
     Service type: DATA_FAX_MODEM
     Service mode: DATA_FAX_MODEM
     Session State: IDLE
  00:02:23: incoming called number: 35160
     Service type: DATA_FAX_MODEM
     Service mode: DATA_FAX_MODEM
     Session State: IDLE
     Service type: DATA_FAX_MODEM
     Service mode: DATA_FAX_MODEM
     Session State: ACTIVE
  00:02:23: Modem State event:
            State: Connect
  00:02:16: Modem State event:
            State: Link
  00:02:13: Modem State event:
            State: Train Up
  00:02:05: Modem State event:
            State: EC Negotiating
  00:02:05: Modem State event:
            State: Steady
  00:02:05: Modem Static event:
    Connect Protocol                        :   LAP-M
    Compression                             :   V.42bis
    Connected Standard                      :   V.34+
    TX,RX Symbol Rate                       :   3429, 3429
    TX,RX Carrier Frequency                 :   1959, 1959
    TX,RX Trellis Coding                    :   16/16
    Frequency Offset                        :   0  Hz
    Round Trip Delay                        :   0  msecs
    TX,RX Bit Rate                          :   33600, 33600
    Robbed Bit Signalling (RBS) pattern     :   0
    Digital Pad                             :   None
    Digital Pad Compensation                :   None
    4 bytes of link info not formatted      :   0x00 0x00 0x00 0x00 0x00
  00:02:06:Modem Dynamic event:
    Sq Value                                :   5
    Signal Noise Ratio                      :   40  dB
    Receive Level                           :   -12  dBm
    Phase Jitter Frequency                  :   0  Hz
```

```
Phase Jitter Level                     :   2  degrees
Far End Echo Level                     :   -90  dBm
Phase Roll                             :   0  degrees
Total Retrains                         :   0
EC Retransmission Count                :   0
Characters transmitted, received       :   0, 0
Characters received BAD                :   0
PPP/SLIP packets transmitted, received  :   0, 0
PPP/SLIP packets received (BAD/ABORTED) :   0
EC packets transmitted, received OK     :   0, 0
EC packets (Received BAD/ABORTED)       :   0
```

Modem Performance Summary Example

You can display a high level summary of the performance of a modem with the **show modem summary** command:

```
Router# show modem summary

         Incoming calls       Outgoing calls      Busied   Failed   No    Succ
Usage  Succ   Fail  Avail   Succ   Fail  Avail    Out      Dial     Ans   Pct.
  14%  2489    123     15      0      0     15        0        3     3     95%
```

Modem AT-Mode Example

The following example shows that modem 1/1 has one open AT directly connected session:

```
Router# show modem at-mode

Active AT-MODE management sessions:
Modem    User's Terminal
1/1 0 cty 0
```

Connection Speed Performance Verification Example

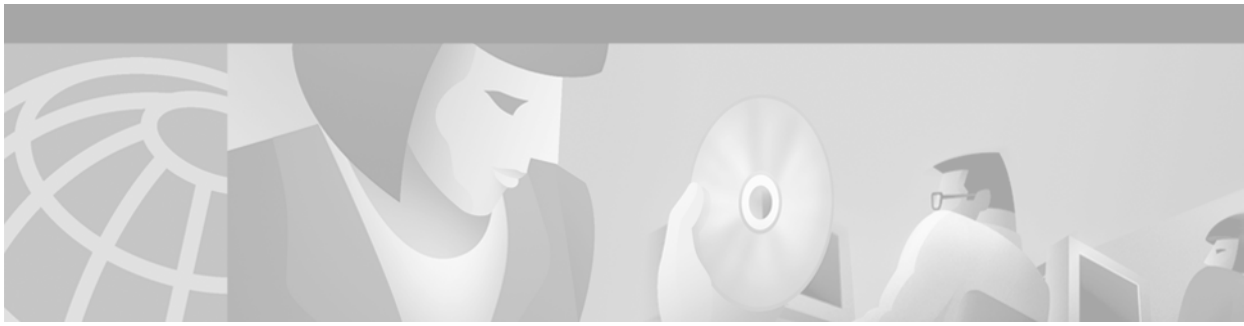
Making sure that your modems are connecting at the correct connection speeds is an important aspect of managing modems. The **show modem connect-speeds** and **show modem** commands provide performance information that allow you to investigate possible inoperable or corrupt modems or T1/E1 lines. For example, suppose you have an access server that is fully populated with V.34 modems. If you notice that modem 1/0 is getting V.34 connections only 50 percent of the time, whereas all the other modems are getting V.34 connections 80 percent of the time, then modem 1/0 is probably malfunctioning. If you are reading low connection speeds across all the modems, you may have a faulty channelized T1 or ISDN PRI line connection.

To display connection speed information for all modems that are running in your system, use the **show modem connect-speeds** *max-speed* EXEC command. Because most terminal screens are not wide enough to display the entire range of connection speeds at one time (for example, 75 to 56,000 bps), the *max-speed* argument is used. This argument specifies the contents of a shifting baud-rate window, which provides you with a snapshot of the modem connection speeds for your system. Replace the *max-speed* argument with the maximum connect speed that you want to display. You can specify from 12,000 to 56,000 bps. If you are interested in viewing a snapshot of lower baud rates, specify a lower connection speed. If you are interested in displaying a snapshot of higher rates, specify a higher connection speed.

The following example displays connection speed information for modems running up to 33,600 bps:

```
Router# show modem connect-speeds 33600

  transmit connect speeds

  Mdm   14400  16800  19200  21600  24000  26400  28800  31200  33600 TotCnt
* 0/0       0      0      0      0      0      0      4      4      1      9
* 0/1       2      0      0      0      0      0      3      3      1      9
  0/2       2      0      0      0      0      1      2      4      1     10
* 0/3       0      0      0      1      0      0      3      4      1      9
* 0/4       1      0      0      0      0      2      2      1      1      7
* 0/5       0      0      0      0      0      0      4      4      1      9
* 0/6       0      0      0      0      0      1      3      3      1      8
* 0/7       0      0      0      2      0      0      4      3      1     10
* 0/8       2      0      0      0      0      0      3      4      1     10
* 0/9       0      0      0      0      0      0      4      3      0      7
* 0/10      1      0      0      0      0      1      3      2      1      8
* 0/11      0      0      0      0      0      0      4      3      1      8
  0/12      1      0      0      0      0      0      4      2      1      8
* 0/13      0      0      0      0      0      0      4      2      1      7
* 0/14      1      0      0      0      0      1      2      2      1      7
* 0/15      0      0      0      0      0      0      4      2      1      7
* 0/16      0      0      0      1      0      0      3      2      1      7
* 0/17      1      0      0      0      0      0      4      2      1      8
* 0/18      1      0      0      0      0      0      3      3      1      8
* 0/19      0      0      0      0      0      0      5      3      1      9
* 0/20      0      0      0      0      0      0      4      2      1      7
* 0/21      1      0      0      0      0      0      4      2      0      7
* 0/22      0      0      0      0      0      0      7      9      1     17
* 0/23      0      0      0      0      0      2      2      3      1      8
* 2/0       0      0      0      1      0      0      3      3      1      8
* 2/1       0      0      0      0      0      0      5      2      1      8
* 2/2       0      0      0      1      0      0      4      1      1      7
* 2/3       1      0      0      0      0      0      4      2      1      8
* 2/4       0      0      0      0      0      0      5      2      1      8
* 2/5       0      0      0      0      0      0      4      3      1      8
* 2/6       0      0      0      0      0      0      3      2      1      6
* 2/7       1      0      0      0      0      1      3      2      0      7
* 2/8       1      0      0      0      0      0      3      2      1      7
* 2/9       0      0      0      0      0      1      3      2      1      7
* 2/10      2      0      0      0      0      2      1      0      1      6
* 2/11      0      0      0      1      0      1      3      5      1     11
* 2/12      0      0      0      0      0      0      5      2      1      8
* 2/13      1      0      0      0      0      0      5      0      1      7
* 2/14      1      0      0      0      0      0      3      3      1      8
* 2/15      1      0      0      0      0      1      2      3      1      8
* 2/16      0      0      0      0      0      0      4      3      1      8
* 2/17      0      0      0      0      0      0      5     11      0     16
* 2/18      0      0      0      1      0      1      1      2      1      6
* 2/19      0      0      0      0      0      0      2      3      1      6
* 2/20      1      0      0      0      0      2      3      9      1     16
* 2/21      1      0      0      0      0      0      4      1      1      7
* 2/22      0      0      0      1      0      0      2      3      1      7
* 2/23      0      0      0      0      0      1      3      3      1      8
 Tot       23      0      0      9      0     18    165    141     44    400
 Tot %      5      0      0      2      0      4     41     35     11

  receive connect speeds

  Mdm   14400  16800  19200  21600  24000  26400  28800  31200  33600 TotCnt
* 0/0       0      0      0      0      0      4      1      3      1      9
* 0/1       2      0      0      0      0      3      1      2      1      9
  0/2       2      0      0      0      0      3      1      3      1     10
```

*	0/3	0	0	0	1	0	3	4	0	1	9
*	0/4	1	0	0	0	0	4	0	1	1	7
*	0/5	0	0	0	0	0	4	3	1	1	9
*	0/6	0	0	0	0	0	4	0	3	1	8
*	0/7	0	0	0	2	0	4	1	2	1	10
*	0/8	2	0	0	0	0	3	0	5	0	10
*	0/9	0	0	0	0	0	4	2	0	1	7
*	0/10	1	0	0	0	0	4	0	2	1	8
*	0/11	0	0	0	0	0	4	0	3	1	8
	0/12	1	0	0	0	0	2	2	2	1	8
*	0/13	0	0	0	0	0	4	1	1	1	7
*	0/14	1	0	0	0	0	2	3	0	1	7
*	0/15	0	0	0	0	0	4	1	1	1	7
*	0/16	0	0	0	1	0	3	2	0	1	7
*	0/17	1	0	0	0	0	4	1	1	1	8
*	0/18	1	0	0	0	0	3	2	1	1	8
*	0/19	0	0	0	0	0	5	1	2	1	9
*	0/20	0	0	0	0	0	4	0	3	0	7
*	0/21	1	0	0	0	0	4	0	1	1	7
*	0/22	0	0	0	0	0	6	6	4	1	17
*	0/23	0	0	0	0	0	4	2	1	1	8
*	2/0	0	0	0	1	0	3	1	2	1	8
*	2/1	0	0	0	0	0	3	3	1	1	8
*	2/2	0	0	0	1	0	4	0	1	1	7
*	2/3	1	0	0	0	0	3	2	1	1	8
*	2/4	0	0	0	0	0	4	2	1	1	8
*	2/5	0	0	0	0	0	4	1	2	1	8
*	2/6	0	0	0	0	0	3	0	3	0	6
*	2/7	1	0	0	0	1	2	2	0	1	7
*	2/8	1	0	0	0	0	3	0	2	1	7
*	2/9	0	0	0	0	0	4	1	1	1	7
*	2/10	2	0	0	0	0	3	0	0	1	6
*	2/11	0	0	0	1	0	3	1	5	1	11
*	2/12	0	0	0	0	0	4	3	0	1	8
*	2/13	1	0	0	0	0	2	3	0	1	7
*	2/14	1	0	0	0	0	3	2	1	1	8
*	2/15	1	0	0	0	0	3	0	3	1	8
*	2/16	0	0	0	0	0	4	0	4	0	8
*	2/17	0	0	0	0	0	5	2	8	1	16
*	2/18	0	0	1	0	0	2	1	1	1	6
*	2/19	0	0	0	0	0	2	2	1	1	6
*	2/20	1	0	0	0	0	4	2	8	1	16
*	2/21	1	0	0	0	0	4	0	1	1	7
*	2/22	0	0	1	0	0	2	0	3	1	7
*	2/23	0	0	0	0	0	4	2	1	1	8
	Tot	23	0	2	7	1	167	64	92	44	400
	Tot %	5	0	0	1	0	41	16	23	11	



Configuring and Managing Cisco Access Servers and Dial Shelves

This chapter describes configuration and monitoring tasks for the Cisco AS5800 and AS5400 access servers, including dial shelves and dial shelf controllers on the Cisco AS5800 access servers in the following main sections:

- Cisco AS5800 Dial Shelf Architecture and DSIP Overview
- How to Configure Dial Shelves
- Port Management Services on Cisco Access Servers
- Upgrading and Configuring SPE Firmware

For further information and configuration examples for the Cisco AS5400, refer to the *Cisco AS5400 Universal Access Server Software Configuration Guide*.

For further information and configuration examples for the Cisco AS5800, refer to the *Cisco AS5800 Universal Access Server Operations, Administration, Maintenance, and Provisioning Guide*.

For more information on the Cisco access servers, go to the Cisco Connection Documentation site on Cisco.com, or use the Cisco Documentation CD-ROM.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

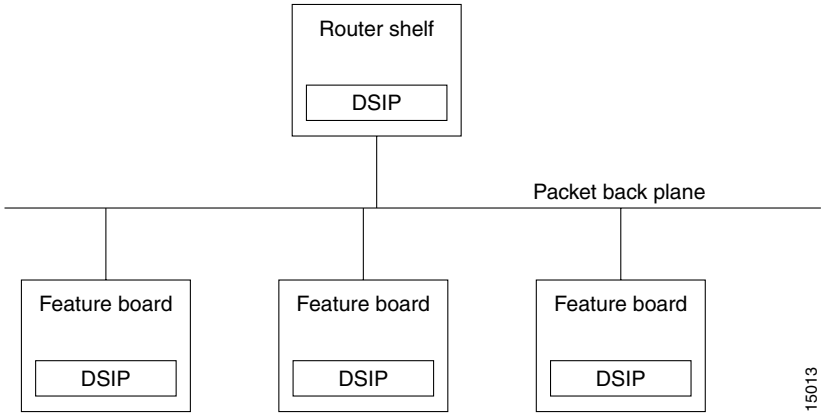
Cisco AS5800 Dial Shelf Architecture and DSIP Overview

The Cisco AS5800 is a rack-mounted system consisting of a router shelf and a dial shelf. The dial shelf contains feature and controller cards (trunk cards), modem cards, and dial shelf controller (DSC) cards.

Note For more information about split dial shelf configuration, refer to the hardware installation guides that accompanied your Cisco AS5800 Universal Access Server and the *Cisco AS5800 Universal Access Server Software Installation and Configuration Guide*.

The Dial Shelf Interconnect Protocol (DSIP) is used for communication between router shelf and dial shelf on an AS5800. Figure 22 diagrams the components of the architecture. The router shelf is the host for DSIP commands, which can be run remotely on the feature boards of the dial shelf using the command, **execute-on**. DSIP communicates over the packet backplane via the dial shelf interconnect (DSI) cable.

Figure 22 DSIP Architecture in the Cisco AS5800



Split Dial Shelves Feature

The split dial shelves feature provides for doubling the throughput of the Cisco AS5800 access server by splitting the dial shelf slots between two router shelves, each router connected to one Dial Shelf Controller (DSC), two of which must be installed in the system. Each router shelf is configured to control a certain set from the range of the dial shelf slots. Each router shelf will operate as though any other slots in the dial shelf contained no cards, even if there is a card in them, because they are controlled by the other router shelf. Thus the configuration on each router shelf would affect only the "owned" slots.

Each router shelf should own modem cards and trunk cards. Calls received on a trunk card belonging to one router shelf cannot be serviced by a modem card belonging to the other router shelf. Each router shelf operates like a single Cisco AS5800 access server system, as if some slots are unavailable.

Refer to the section "Configuring Dial Shelf Split Mode" for more information about configuring split dial shelves.

How to Configure Dial Shelves

To configure and maintain dial shelves, perform the tasks in the following sections:

- Configuring the Shelf ID
- Configuring Redundant DSC Cards
- Synchronizing to the System Clocks
- Configuring Dial Shelf Split Mode
- Executing Commands Remotely
- Verifying DSC Configuration

- Monitoring and Maintaining the DSCs
- Troubleshooting DSIP

Configuring the Shelf ID

The Cisco AS5800 consists of a router shelf and a dial shelf. To distinguish the slot/port number on the Cisco AS5800, you must specify the shelf number. The default shelf number is 0 for the router shelf and 1 for the dial shelf.

Caution You must reload the Cisco AS5800 for the new shelf number to take effect. Because the shelf number is part of the interface names when you reload, all NVRAM interface configuration information is lost.

Normally you do not need to change the shelf IDs; however, if you do, we recommend that you change the shelf number when you initially access the setup facility. For information on the setup facility, refer to the *Cisco AS5800 Universal Access Server Software Installation and Configuration Guide*.

If you are booting the router shelf from the network (netbooting), you can change the shelf numbers using the **shelf-id** command.

To configure the dial shelf, you save and verify the configuration in EXEC mode, and enter **shelf-id** commands in global configuration mode, as indicated in the following steps:

	Command	Purpose
Step 1	`Router# copy startup-configure tftp`	Saves your current configuration. Changing the shelf number removes all interface configuration information when you reload the Cisco AS5800.
Step 2	`Router# configure terminal`	Begins global configuration mode.
Step 3	`Router(config)# shelf-id` *number* `router-shelf`	Specifies the router shelf ID.
Step 4	`Router(config)# shelf-id` *number* `dial-shelf`	Specifies the dial shelf ID.
Step 5	`Router(config)# exit`	Exits global configuration mode.
Step 6	`Router# copy running-config startup-config`	Saves your configuration. This step is optional.
Step 7	`Router# show version`	Verifies that the correct shelf number will be changed after the next reload.
Step 8	`Router# reload components all`	Instructs the DSC (or DSCs in a redundant configuration) be reloaded at the same time as a reload on the router shelf. Type “yes” to the “save config” prompt. Configure one interface so that its router shelf has connectivity to the server with the configuration.
Step 9	`Router# copy tftp startup-config`	Because changing the shelf number removes all interface configuration information when you reload the Cisco AS5800, edit the configuration file saved in step 1 and download it.

If you are booting the router shelf from Flash memory, use the following commands beginning in EXEC mode:

	Command	Purpose

Step 1	`Router# copy running-config tftp`
	or
	`Router# copy startup-config tftp`
		Saves your current (latest) configuration to a server.

Step 2	`Router# configure terminal`	Begins global configuration mode.
Step 3	`Router(config)# shelf-id` *number* `router-shelf`	Configures the router shelf ID.
Step 4	`Router(config)# shelf-id` *number* `dial-shelf`	Configures the dial shelf ID.
Step 5	`Router(config)# exit`	Exits global configuration mode.
Step 6	`Router> copy running-config startup-config`	Saves your configuration. This step is optional. If this step is skipped, type "No" at the "save configuration" prompt.
Step 7	`Router> show version`	Allows verification that the correct shelf number will be changed after the next reload. Edit the configuration file saved in Step 1.
Step 8	`Router> copy tftp startup-config`	Copies the edited configuration to NVRAM on the Cisco AS5800.
Step 9	`Router# reload components all`	Instructs the DSC (or DSCs in a redundant configuration) to be reloaded at the same time as a reload on the router shelf.

Configuring Redundant DSC Cards

The Redundant Dial Shelf Controller feature consists of two DSC cards on a Cisco AS5800 dial shelf. The DSC cards provide clock and power control to the dial shelf cards. Each DSC card provides the following:

- Master clock for the dial shelf
- Fast Ethernet link to the router shelf
- Environmental monitoring of the feature boards
- Bootstrap images on start-up for the feature boards

The Redundant Dial Shelf Controller feature is automatically enabled when two DSC cards are installed. DSC redundancy is supported with Cisco AS5800 software at the Dial Shelf Interconnect Protocol (DSIP) level.

This feature enables a Cisco AS5800 dial shelf to use dual DSCs for full redundancy. A redundant configuration allows for one DSC to act as backup to the active card, should the active card fail. This increases system availability by preventing loss of service. The redundant DSC functionality is robust under high loads and through DSC or software crashes and reloads. The redundant DSC functionality is driven by the following events:

- User actions
- Control messages
- Timeouts

- Detection of component failures
- Error and warning messages

DSC redundancy provides maximum system availability by preventing loss of service if one of the DSCs fails. There is no load sharing between the Broadband Inter-Carrier Interfaces (BICI). One BIC is used as a backup, carrying only control traffic, such as keepalives, until there is a switchover.

Before starting this configuration task:

- Your Cisco AS5800 router shelf and dial shelf must be fully installed, with two DSC cards installed on the dial shelf.
- Your Cisco AS5800 access server must be running Cisco IOS Release 12.1(2)T.
- The external DSC clocking port must be configured identically on both router shelves and must be physically connected to both DSCs. This assures that if a DSC card needs replacing or if the backup DSC card becomes primary, clocking remains stable.

Synchronizing to the System Clocks

The time-division multiplexing (TDM) bus in the backplane on the dial shelf must be synchronized to the T1/E1 clocks on the trunk cards. The Dial Shelf Controller (DSC) card on the daily shelf provides hardware logic to accept multiple clock sources as input and use one of them as the primary source to generate a stable, PPL synchronized output clock. The input clock can be any of the following sources:

- Trunk port in slots 0 through 5—up to 12 can be selected (2 per slot)
- An external T1 or E1 clock source fed directly through a connector on the DSC card
- A free-running clock from an oscillator in the clocking hardware on the DSC card

For dual (redundant) DSC cards, the external DSC clocking port should be configured so that the clock signal fed into both DSCs is identical.

To configure the external clocks, use the following commands from the router shelf login beginning in global configuration mode. One external clock is configured as the primary clock source, and the other is configured as the backup clock source.

	Command	Purpose
Step 1	`Router(config)# dial-tdm-clock priority value`	Configures the trunk card clock priority. Priority range is a value between 1 and 50.
Step 2	`Router(config)# dial-tdm-clock priority X {trunk-slot Y port Z} external {t1 | e1} [120-ohm]`	Selects the T1/E1 trunk slot and port that is providing the clocking source. T1/E1 selection is based on the incoming signal. Select the impedance. The default impedance is 75-ohm.

Step 3	`Router(config)# dial-tdm-clock priority value external t1`
	or
	`Router(config)# dial-tdm-clock priority value external e1`
		Configures the T1/E1 external clock on the dial shelf controller front panel. T1/E1 selection is based on the signal coming in. Priority range is a value between 1 and 50.

Step 4	`Router(config)# Ctrl-Z` `Router#`	Verifies your command registers when you press the return key. Enter **Ctrl-Z** to return to privileged EXEC mode.
Step 5	`Router# copy running-config startup-config`	Saves your changes.

Verifying External Clock Configuration

To verify that the primary clock is running, enter the **show dial-shelf clocks** privileged EXEC command:

```
Router# show dial-shelf 12 clocks

Slot 12:
System primary is 1/2/0 of priority 202
TDM Bus Master Clock Generator State = NORMAL
Backup clocks:
Source  Slot    Port    Priority      Status      State
------------------------------------------------------
Trunk   2       1       208            Good        Default
Slot    Type    11 10  9  8  7  6  5  4  3  2  1  0
2       T1       G  G  G  G  G  G  G  G  G  G  G  G
```

For more information on configuring external clocks, refer to the Cisco document *Managing Dial Shelves.*

Configuring Dial Shelf Split Mode

This section describes the procedure required to transition a router from normal mode to split mode and to change the set of slots a router owns while it is in split mode. Since the process of switching the ownership of a slot from one router to the other is potentially disruptive (when a feature board is restarted, all calls through that card are lost), a router shelf cannot take over a slot until ownership is relinquished by the router that currently claims ownership, either by reconfiguring the router or disconnecting that router or its associated DSC.

The dial shelf is split by dividing the ownership of the feature boards between the two router shelves. You must configure the division of the dial shelf slots between the two router shelves so that each router controls an appropriate mix of trunk and modem cards. Each router shelf controls its set of feature boards as if those were the only boards present. There is no interaction between feature boards owned by one router and feature boards owned by the other router.

Split mode is entered when the **dial-shelf split slots** command is parsed on the router shelf. This can occur when the router is starting up and parsing the stored configuration, or when the command is entered when the router is already up. Upon parsing the **dial-shelf split slots** command, the router frees any resources associated with cards in the slots that it no longer owns, as specified by exclusion of slot numbers from the *slot-numbers* argument. The router should be in the same state as if the card had been removed from the slot; all calls through that card will be terminated. The configured router then informs its connected DSC that it is in split mode, and which slots it claims to own.

In split mode, a router shelf by default takes half of the 2048 available TDM timeslots. The TDM split mode is configured using the **dial-shelf split backplane-ds0** command. (The **dial-shelf split slot** command must be defined for the **dial-shelf split backplane-ds0** command to be active.) If the **dial-shelf split slots** command is entered when the total number of calls using timeslots exceeds the number that would normally be available to the router in split mode, the command is rejected. This should occur only when a change to split mode is attempted, in which the dial shelf has more than 896 calls in progress (more than half of the 1,792 available timeslots). Otherwise, a transition from normal mode to split mode can be made without disturbing the cards in the slots that remain owned, and calls going through those cards will stay up.

To configure a router for split dial shelf operation, perform the following steps:

Step 1 Ensure that both DSCs and both router shelves are running the same Cisco IOS image.

Note Having the same version of Cisco IOS running on both DSCs and both router shelves is not mandatory; however, it is a good idea. There is no automatic checking that the versions are the same.

Step 2 Schedule a time when the Cisco AS5800 can be taken out of service without unnecessarily terminating calls in progress. The entire procedure for transitioning from normal mode to split mode should require approximately one hour if all the hardware is already installed.

Step 3 Busy out all feature boards and wait for your customers to log off.

Step 4 Reconfigure the existing router shelf to operate in split mode.

Step 5 Enter the **dial-shelf split slots** command, specifying the slot numbers that are to be owned by the existing router shelf.

Step 6 Configure the new router shelf to operate in split mode on other feature boards.

Step 7 Enter the **dial-shelf split slots** command, specifying the slot numbers that are to be owned by the new router shelf. Do not specify any of the slot numbers that you specified in Step 6. The range of valid slot numbers is 0 through 11.

To perform this step, enter the following command in global configuration mode:

Command	Purpose
`Router(config)# dial-shelf split slots` *slot-numbers*	Enter list of slot numbers, for example: `dial-shelf split slots 0 1 2 6 7 8` In this example, the other router shelf could be configured to own the other slots: 3 4 5 9 10 11. **Normal mode**: This command changes the router shelf to split mode with ownership of the slots listed. In case of conflicting slot assignments, the command is rejected and a warning message is issued. Issue a **show dial-shelf split slots** command to the other router shelf to display its list of owned dial shelf slots. Online insertion and removal (OIR) events on all slots are detected by both DSCs and added to the list of feature boards physically present in the dial shelf; however, OIR event processing is done only for assigned slots. **Split mode**: This command adds the dial shelf slots listed to the router shelf's list of owned dial shelf slots.

Step 8 Install the second DSC, if it has not already been installed.

Step 9 Connect the DSIP cable from the second DSC to the new router shelf.

Step 10 Ensure that split mode is operating properly.

Enter the **show dial-shelf** command for each router. This command has been extended so that the response indicates that the router shelf is running in split mode and which slots the router shelf owns. The status of any cards in any owned slots is shown, just as they are in the present **show dial-shelf** command. When in split mode, the output will be extended as in the following example:

```
System is in split dial shelf mode.
Slots owned: 0 2 3 4 5 6 (connected to DSC in slot 13)
Slot    Board     CPU       DRAM          I/O Memory   State   Elapsed
         Type     Util    Total (free)   Total (free)          Time
 0        CE1    0%/0%  21341728( 87%)  8388608( 45%)  Up      00:11:37
 2        CE1    0%/0%  21341728( 87%)  8388608( 45%)  Up      00:11:37
 4 Modem(HMM)  20%/20%   6661664( 47%)  6291456( 33%)  Up      00:11:37
 5 Modem(DMM)    0%/0%   6661664( 31%)  6291456( 32%)  Up      00:11:37
 6 Modem(DMM)    0%/0%   6661664( 31%)  6291456( 32%)  Up      00:11:37
13        DSC    0%/0%  20451808( 91%)  8388608( 66%)  Up      00:16:31
Dial shelf set for auto boot
```

Step 11 Enable all feature boards to accept calls once again.

Changing Slot Sets

You can change the sets of slots owned by the two router shelves while they are in split mode by first removing slots from the set owned by one router, and then adding them to the slot set of the other router. The changed slot set information is sent to the respective DSCs, and the DSCs determine which slots have been removed and which added from the new slot set information. It should be clear that moving a slot in this manner will disconnect all calls that were going through the card in that slot.

To perform this task, enter the following commands as needed:

Command	Purpose
`Router (config)# dial-shelf split slots remove` *slot-numbers*	Removes the dial shelf slots listed from the router shelf's list of owned dial shelf slots. The effect of multiple commands is cumulative.
`Router(config)# dial-shelf split slots` *slot-numbers*	Adds the dial shelf slots listed to the router shelf's list of owned dial shelf slots.

When a Slot Is Removed

The router shelf that is losing the slot frees any resources and clears any state associated with the card in the slot it is relinquishing. The DSC reconfigures its hub to ignore traffic from that slot, and if there is a card in the slot, it will be reset. This ensures that the card frees up any TDM resource it might be using and allows it to restart under control of the router shelf that is subsequently configured to own the slot.

When a Slot Is Added

If there are no configuration conflicts, and there is a card present in the added slot, a dial-shelf OIR insertion event is sent to the router shelf, which processes the event the same as it always does. The card in the added slot is reset by the DSC to ensure a clean state, and the card downloads its image from the router shelf that now owns it.

If the other router shelf and the other DSC claim ownership of the same slot, the command adding the slot should be rejected. However, should a configuration conflict exist, error messages are sent to both routers and the card is not reset until one of the other router shelves and its DSC stop claiming ownership of the slot. Normally, this will not happen until you issue a **dial-shelf split slots remove** command surrendering the ownership claim on the slot by one of the routers.

Leaving Split Mode

Split mode is exited when the dial shelf configuration is changed by a **no dial-shelf split slots** command. When the split dial shelf line is removed, the router shelf will start using all of the TDM timeslots. Feature boards that were not owned in split mode and that are not owned by the other router will be reset. Cards in slots that are owned by the other router will be reset, but only after the other DSC has been removed or is no longer claiming the slots. The split dial shelf configuration should not be removed while the second router shelf is still connected to the dial shelf.

When a router configured in split mode fails, all calls associated with the failed router are lost. Users cannot connect back in until the failed router recovers and is available to accept new incoming calls; however, the other split mode router shelf will continue to operate normally.

Troubleshooting Split Dial Shelves

The system will behave as configured as soon as the configuration is changed. The exception is when there is a misconfiguration, such as when one router is configured in split mode and the other router is configured in normal mode, or when both routers are configured in split mode and both claim ownership of the same slots.

Problems can arise if one of the two routers connected to a dial shelf is not configured in split mode, or if both are configured in split mode and both claim ownership of the same slots. If the state of the second router is known when the **dial-shelf split slots** command is entered and the command would result in a conflict, the command is rejected.

If a conflict in slot ownership does arise, both routers will receive warning messages until the conflict is resolved. Any card in a slot which is claimed by both routers remains under the control of the router that claimed it first, until you can resolve the conflict by correcting the configuration of one or both routers.

It should be noted that there can also be slots that are not owned by either router (orphan slots). Cards in orphan slots cannot boot up until one of the two routers claims ownership of the slot because neither DSC will download bootstrap images to cards in unowned orphan slots.

Managing a Split Dial Shelf

If you are installing split dial shelf systems, a system controller is available that provides a single system view of multiple point of presences (POPs). The system controller for the Cisco AS5800 Universal Access Server includes the Cisco 3640 router running Cisco IOS software. The system controller can be installed at a remote facility so that you can access multiple systems through a console port or Web interface.

There are no new MIBs or MIB variables required for the split dial shelf configuration. A split dial shelf appears to Simple Network Management Protocol (SNMP) management applications as two separate Cisco AS5800 systems. One console to manage the whole system is not supported—you must have a console session per router shelf (two console sessions) to configure each split of the Cisco AS5800. The system controller must manage a split dial shelf configuration as two separate Cisco AS5800 systems.

The normal mode configuration of the Cisco AS5800 requires the dial shelf and router shelf IDs to be different. In a split system, four unique shelf IDs are desirable, one for each router shelf and one for each of the slot sets; however, a split system will function satisfactorily if the router shelf IDs are the same. If a system controller is used to manage a split dial shelf configuration, the two routers must have distinct shelf IDs, just as they must when each router has its own dial shelf.

You can download software configurations to any Cisco AS5800 using SNMP or a Telnet connection. The system controller also provides performance monitoring and accounting data collection and logging.

In addition to the system controller, a network management system with a graphical user interface (GUI) runs on a UNIX SPARC station and includes a database management system, polling engine, trap management, and map integration.

To manage a split dial shelf, enter the following commands in EXEC mode as needed:

Command	Purpose
`Router#` **`show dial-shelf split`**	Displays the slots assigned to each of the router shelves and the corresponding feature boards in 'orphan' slots (slots not currently assigned to either router).
`Router#` **`show dial-shelf`**	Displays information about the dial shelf, including clocking information.
`Router#` **`show context`**	Displays information about the dial shelf, including clocking information, but works only for owned slots. Use **show context all** to display all the information available about any slot. This is intended to cover the case where ownership of a feature board is moved from one router shelf to the other after a crash.

Executing Commands Remotely

Although not recommended, it is possible to connect directly to the system console interface in the DSC to execute dial shelf configuration commands. All commands necessary for dial shelf configuration, and **show**, and **debug** command tasks can be executed remotely from the router console. A special command, **execute-on**, is provided for this purpose. This command enables a special set of EXEC mode commands to be executed on the router or the dial shelf. This command is a convenience that avoids connecting the console to the DSC. For a list of commands you can execute using **execute-on**, refer to the command description in the *Cisco IOS Dial Technologies Command Reference*.

To enter a command that you wish to execute on a specific card installed in the dial shelf while logged onto the router shelf console, use the following commands in privileged EXEC mode as needed:

Command	Purpose
`Router#` **`execute-on slot`** *`slot command`*	Executes a command from the router shelf on a specific slot in the dial shelf.
`Router#` **`execute-on all`** *`command`*	Executes a command from the router shelf on all cards in the dial shelf.

Verifying DSC Configuration

To verify that you have started the redundant DSC feature, enter the **show redundancy** privileged EXEC command:

```
Router# show redundancy

DSC in slot 12:

Hub is in 'active' state.
Clock is in 'active' state.

DSC in slot 13:

Hub is in 'backup' state.
Clock is in 'backup' state.

Router#
```

Monitoring and Maintaining the DSCs

To monitor and maintain the DSC cards, use the following commands in privileged EXEC mode, as needed:

Command	Purpose
Router# **hw-module** *shelf/slot* {**start**|**stop**}	Stops the target DSC remotely from the router console. Restart the DSC if it has been stopped.
Router# **show redundancy** [**history**]	Displays the current or history status for redundant DSC.
Router# **debug redundancy** {**all**|**ui**|**clk**|**hub**}	Use this debug command if you need to collect events for troubleshooting, selecting the appropriate required key word.
Router# **show debugging**	Lists the debug commands that are turned on, including those for redundant DSC.

Troubleshooting DSIP

There are a number of show commands available to aid in troubleshooting dial shelves. Use the following EXEC mode commands to monitor DSI and DSIP activity as needed:

Command	Purpose
Router# **clear dsip tracing**	Clears tracing statistics for the DSIP.
Router# **show dsip**	Displays all information about the DSIP.
Router# **show dsip clients**	Displays information about DSIP clients.
Router# **show dsip nodes**	Displays information about the processors running the DSIP.
Router# **show dsip ports**	Displays information about local and remote ports.
Router# **show dsip queue**	Displays the number of messages in the retransmit queue waiting for acknowledgment.
Router# **show dsip tracing**	Displays DSIP tracing buffer information.

Command	Purpose
`Router# show dsip transport`	Displays information about the DSIP transport statistics for the control/data and IPC packets and registered addresses.
`Router# show dsip version`	Displays DSIP version information.

The privileged EXEC mode **show dsi** command can also be used to troubleshoot, as it displays the status of the DSI adapter, which is used to physically connect the router shelf and the dial shelf to enable DSIP communications.

The following is an example troubleshooting scenario:

Problem: The router shelf boots, but there is no communication between the router and dial shelves.

Step 1 Run the **show dsip transport** command.

Step 2 Check the "DSIP registered addresses" column. If there are zero entries there, there is some problem with the Dial Shelf Interconnect (DSI). Check if the DSI is installed in the router shelf.

Step 3 If there is only one entry and it is our own local address, then first sanity check the physical layer. Make sure that there is a physical connection between the RS and DS. If everything is fine from cabling point of view, go to step 3.

Step 4 Check the DSI health by issuing the **show dsi** command. This gives a consolidated output of DSI controller and interface. Check for any errors like runts, giants, throttles and other usual FE interface errors.

Diagnosis: If an entry for a particular dial shelf slot is not found among the registered addresses, but most of other card entries are present, the problem is most likely with that dial shelf slot. The DSI hardware on that feature board is probably bad.

Port Management Services on Cisco Access Servers

Port Management Services on the Cisco AS5400 Access Server

Port service management on the Cisco AS5400 access server implements service using the NextPort dial feature card (DFC). The NextPort DFC is a hardware card that processes digital service port technology for the Cisco AS5400 access server. A port is defined as an endpoint on a DFC card through which multiservice tones and data flow. The ports on the NextPort DFC support both modem and digital services. Ports can be addressed-aggregated at the slot level of the NextPort module, the Service Processing Element (SPE) level within the NextPort module, and the individual port level. Cisco IOS Release 12.1(3)T or higher is required for the NextPort DFC.

Instead of the traditional line-modem one-to-one correspondence, lines are mapped to an SPE that resides on the Cisco AS5400 NextPort DFC. Each SPE provides modem services for six ports. Busyout and shutdown can be configured at the SPE or port level. The NextPort DFC introduces the slot and SPE software hierarchy. On the Cisco AS5400, the hierarchy designation is *slot*/*SPE*.

The NextPort DFC slot is defined as a value between 1 and 7. Slot 0 is reserved for the motherboard. Each NextPort DFC provides 18 SPEs. The SPE value ranges from 0 to 17. Since each SPE has six ports, the NextPort DFC has a total of 108 ports. The port value ranges from 0 to 107.

The NextPort DFC performs the following functions:

- Converts pulse code modulation (PCM) bitstreams to digital packet data.
- Forwards converted and packetized data to the main processor, which examines the data and forwards it to the backhaul egress interface.
- Supports all modem standards (such as V.34 and V.42*bis*) and features, including dial-in and dial-out.

Port Management Services on the Cisco AS5800 Access Server

Port service management on the Cisco AS5800 access server implements service on the Universal Port Card (UPC). A universal port carries a single channel at the speed of digital signal level 0 (DS0), or the equivalent of 64-kbps on a T1 facility.

Network traffic can be a modem, voice, or fax connection. The 324 port UPC uses NextPort hardware and firmware to provide universal ports for the Cisco AS5800 access server. These ports are grouped into 54 service processing elements (SPEs). Each SPE supports six universal ports. To find the total number of ports supported by a UPC, multiply the 54 SPEs by the six ports supported on each SPE. The total number of universal ports supported by a single UPC is 324. Configuration, management, and troubleshooting of universal ports can be done at the UPC, SPE, and port level. Each UPC also has a SDRAM card with a minimum of a 128 MB of memory.

The Cisco AS5800 access server can be equipped with a maximum of seven UPCs with upgradable firmware. The UPC supports data traffic, and depending on the software and platform is universal port capable. Each UPC plugs directly into the dial shelf backplane and does not need any external connections. Each UPC has three LEDs, which indicate card status.

The Cisco AS5800 access server is capable of terminating up to 2,048 incoming modem connections (slightly more than an OC3) when equipped with seven UPCs and three CT3 trunk cards. A split shelf configuration with a second router shelf and second dial shelf controller are required to achieve full capacity. A single router with a standard configuration supports up to 1,344 port connections. Cisco IOS Release 12.1(3)T or higher is required for the UPC. Unless your system shipped with UPCs installed, you must upgrade the Cisco IOS image on the dial shelf and router shelf or shelves.

Instead of the traditional line-modem one-to-one correspondence, lines are mapped to an SPE that resides on the Cisco AS5800 access server UPC. Each SPE provides modem services for six ports. Busyout and shutdown can be configured at the SPE or port level. The UPC introduces the shelf, slot, and SPE software hierarchy. On the Cisco AS5800 access server, the hierarchy designation is *shelf*/*slot*/*SPE*.

A UPC can be installed in slots numbered 2 to 11 on the dial shelf backplane. If installed in slots 0 or 1, the UPC automatically powers down. Slots 0 and 1 only accept trunk cards; they do not accept mixes of cards. We recommend that you install mixes of T3 and T1 cards, or E1 trunk cards in slots 2 to 5. You can use double-density modem cards, UPCs, and VoIP cards simultaneously. Trunk cards can operate in slots 0 to 5 and are required for call termination.

The UPC performs the following functions:

- Converts pulse code modulation (PCM) bitstreams to digital packet data.
- Forwards converted and packetized data to the dial shelf main processor, which examines the data and forwards it to the router shelf. From the router shelf, the data is routed to the external network.

- Supports all modem standards (such as V.34 and V.42*bis*) and features, including dial-in and dial-out.
- Supports online insertion and removal (OIR), a feature that allows you to remove and replace UPCs while the system is operating. A UPC can be removed without disrupting the operation of other cards and their associated calls. If a UPC is removed while the system is operating, connections or current calls on that card are dropped. Calls being handled by other cards are not affected.

Note All six ports on an SPE run the same firmware.

Upgrading and Configuring SPE Firmware

SPE firmware is automatically downloaded in both the Cisco AS5400 and AS5800 access servers.

AS5400 Access Server

SPE firmware is automatically downloaded to a NextPort DFC from the Cisco AS5400 when you boot the system for the first time, or when you insert a NextPort DFC while the system is operating. When you insert DFCs while the system is operating, the Cisco IOS image recognizes the cards and downloads the required firmware to the cards.

The SPE firmware image is bundled with the access server Cisco IOS image. The SPE firmware image uses an *autodetect* mechanism, which enables the NextPort DFC to service multiple call types. An SPE detects the call type and automatically configures itself for that operation. For further information on upgrading SPE firmware from the Cisco IOS image, refer to the section "Configuring SPEs to Use an Upgraded Firmware File."

The firmware is upgradeable independent of Cisco IOS upgrades, and different firmware versions can be configured to run on SPEs in the same NextPort DFC. You can download firmware from the Cisco System Cisco.com File Transfer Protocol (FTP) server.

AS5800 Access Server

SPE firmware is automatically downloaded to an AS5800 UPC from the router shelf Cisco IOS image when you boot the system for the first time or when you insert a UPC while the system is operating. The Cisco IOS image recognizes the card and the dial shelf downloads the required portware to the cards. Cisco IOS Release 12.1(3)T or higher is required for the UPC.

The SPE firmware image (also known as *portware*) is bundled with the Cisco IOS UPC image. The SPE firmware image uses an *autodetect* mechanism, which enables the UPC to service multiple call types. An SPE detects the call type and automatically configures itself for that operation. For further information on upgrading SPE firmware from the Cisco IOS image, refer to the section "Configuring SPEs to Use an Upgraded Firmware File."

The firmware is upgradable independent of Cisco IOS upgrades, and different firmware versions can be configured to run on SPEs in the same UPC. You can download firmware from the Cisco.com File Transfer Protocol (FTP) server.

Firmware Upgrade Task List

Upgrading SPE firmware from the Cisco.com FTP server is done in two steps:

- Downloading SPE Firmware from the Cisco.com FTP Server to a Local TFTP Server
- Copying the SPE Firmware File from the Local TFTP Server to the SPEs

Firmware Configuration Task List

To complete firmware configuration once you have downloaded the SPE firmware, perform the tasks in the following sections:

- Specifying a Country Name
- Configuring Dial Split Shelves (AS5800 Only)
- Configuring SPEs to Use an Upgraded Firmware File
- Disabling SPEs
- Rebooting SPEs
- Configuring Lines
- Configuring Ports
- Verifying SPE Line and Port Configuration
- Configuring SPE Performance Statistics
- Clearing Log Events
- Troubleshooting SPEs
- Monitoring SPE Performance Statistics

Note The following procedure can be used for either a Cisco AS5400 or AS5800 access server.

Downloading SPE Firmware from the Cisco.com FTP Server to a Local TFTP Server

Note You must be a registered Cisco user to log in to the Cisco Software Center.

You can download software from the Cisco Systems Cisco.com FTP server using an Internet browser or using an FTP application. Both procedures are described.

Using an Internet Browser

Step 1 Launch an Internet browser.

Step 2 Bring up the Cisco Software Center home page at the following URL (this is subject to change without notice):

http://www.cisco.com/kobayashi/sw-center/

Step 3 Click **Access Software** (under Cisco Software Products) to open the Access Software window.

Step 4 Click **Cisco AS5400 Series** or **Cisco AS5800 Series** software.

Step 5 Click the SPE firmware you want and download it to your workstation or PC. For example, to download SPE firmware for the universal access server, click **Download Universal Images**.

Step 6 Click the SPE firmware file you want to download, and then follow the remaining download instructions. If you are downloading the SPE firmware file to a PC, make sure that you download the file to the c:/tftpboot directory; otherwise, the download process does not work.

Step 7 When the SPE firmware is downloaded to your workstation, transfer the file to a Trivial File Transfer Protocol (TFTP) server in your LAN using a terminal emulation software application.

Step 8 When the SPE firmware is downloaded to your workstation, transfer the file to a TFTP server somewhere in your LAN using a terminal emulation software application.

Using an FTP Application

Note The directory path leading to the SPE firmware files on cco.cisco.com is subject to change without notice. If you cannot access the files using an FTP application, try the Cisco Systems URL http://www.cisco.com/cgi-bin/ibld/all.pl?i=support&c=3.

Step 1 Log in to the Cisco.com FTP server called cco.cisco.com:

```
terminal> ftp cco.cisco.com
Connected to cio-sys.cisco.com.
```

Step 2 Enter your registered username and password (for example, **harry** and **letmein**):

```
Name (cco.cisco.com:harry): harry
331 Password required for harry.
Password: letmein
230-###############################################################
230-#  Welcome to the Cisco Systems CCO FTP server.
230-#  This server has a number of restrictions.  If you are not familiar
230-#  with these, please first get and read the /README or /README.TXT file.
230-#  http://www.cisco.com/acs/info/cioesd.html for more info.
230-###############################################################
```

Step 3 Specify the directory path that holds the SPE firmware you want to download. For example, the directory path for the Cisco AS5400 SPE firmware is /cisco/access/5400:

```
ftp> cd /cisco/access/5400
250-Please read the file README
250-  it was last modified on Tue May 27 10:07:38 1997 - 48 days ago
250-Please read the file README.txt
250-  it was last modified on Tue May 27 10:07:38 1997 - 48 days ago
250 CWD command successful.
```

Step 4 Enter the **ls** command to view the contents of the directory:

```
ftp> ls
227 Entering Passive Mode (192,31,7,130,218,128)
150 Opening ASCII mode data connection for /bin/ls.
total 2688
drwxr-s--T   2 ftpadmin ftpcio    512 Jun 30 18:11 .
drwxr-sr-t  19 ftpadmin ftpcio    512 Jun 23 10:26 ..
lrwxrwxrwx  1  root     3          10 Aug   6 1996     README ->README.txt
-rw-rw-r--   1 root     ftpcio   2304 May 27 10:07 README.txt
-r--r--r-- 1 ftpadmin ftpint 377112 Jul 10 18:08 np-spe-upw-10.0.1.2.bin
-r--r--r-- 1 ftpadmin ftpint 635 Jul 10 18:08 SPE-firmware.10.1.30.readme
```

Step 5 Specify a binary image transfer:

```
ftp> binary
200 Type set to I.
```

Step 6 Copy the SPE firmware files from the access server to your local environment with the **get** command.

Step 7 Quit your terminal session:

```
ftp> quit
Goodbye.
```

Step 8 Enter the **ls -al** command to verify that you successfully transferred the files to your local directory:

```
server% ls -al
total 596
-r--r--r-- 1 280208 Jul 10 18:08 np-spe-upw-10.0.1.2.bin
server% pwd
/auto/tftpboot
```

Step 9 Transfer these files to a local TFTP or remote copy protocol (RCP) server that your access server or router can access.

Copying the SPE Firmware File from the Local TFTP Server to the SPEs

The procedure for copying the SPE firmware file from your local TFTP server to the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs is a two-step process. First, transfer the SPE firmware to the access server's Flash memory. Then, configure the SPEs to use the upgrade firmware. The upgrade occurs automatically, either as you leave configuration mode, or as specified in the configuration.

These two steps are performed only once. After you copy the SPE firmware file into Flash memory for the first time, you should not have to perform these steps again.

Note Because the SPE firmware is configurable for individual SPEs or ranges of SPEs, the Cisco IOS software automatically copies the SPE firmware to each SPE each time the access server restarts.

To transfer SPE Firmware to Flash memory, perform the following task to download the Universal SPE firmware to Flash memory:

Step 1 Check the image in the access server Flash memory:

```
Router# show flash
System flash directory:
File  Length   Name/status
  1   4530624  c5400-js-mx
 [498776 bytes used, 16278440 available, 16777216 total]
16384K bytes of processor board System flash (Read/Write)
```

Step 2 Enter the **copy tftp flash** command to download the code file from the TFTP server into the access server Flash memory. You are prompted for the download destination and the remote host name.

```
Router# copy tftp flash
```

Step 3 Enter the **show flash** command to verify that the file has been copied into the access server Flash memory:

```
Router# show flash
```

Specifying a Country Name

To set the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs to be operational for call set up, you must specify the country name. To specify the country name, use the following command in global configuration mode:

Command	Purpose
`Router(config)# spe country` *country name*	Specifies the country to set the UPC or DFC parameters (including country code and encoding). If you do not specify a country, the interface uses the default. If the access server is configured with T1 interfaces, the default is **usa**. If the access server is configured with E1 interfaces, the default is **e1-default**. Use the **no** form of this command to set the country code to the default of the domestic country. **Note** All sessions in all UPCs or DFCs in all slots must be in the idle state for this command to execute.

Configuring Dial Split Shelves (AS5800 Only)

The Cisco AS5800 access server requires a split dial shelf configuration using two router shelves to achieve the maximum capacity of 2048 port connections using the seven UPCs and three T3 + 1 T1 trunks. A new configuration command is available to define the split point:

dial-shelf split backplane-ds0 *option*

The options for this command come in pairs, and vary according to the desired configuration. You will need to log in to each router shelf and separately configure the routers for the intended load. In most circumstances it is recommended that the predefined options are selected. These options are designed to be matched pairs as seen below.

	Router Shelf 1			Router Shelf 2			Total
Option Pair	Option	Maximum Calls	Unused T1	Option	Maximum Calls	Unused T1	
1	**2ct3cas**	1344		**1ct3cas**	672		2016
2	**part2ct1ct3cas**	1152	4	**part1ct1ct3cas**	888	3	2040
3	**2ct3isdn**	1288		**part1ct1ct3isdn_b**	644	7	1932
4	**part2ct1ct3isdn**	1150	2	**part1ct1ct3isdn**	897	1	2047
5[1]	**3ce1**	960		**3ce1**	960		1920
6	Default (no option entered)	1/2 of current input		Default (no option entered)	1/2 of current input		
7	**no dial-shelf backplane-ds0**	1024		**no dial-shelf backplane-ds0**	1024		2048

1. This option is used to revert to the default for an environment using 6 E1 lines.

The **dial-shelf split slot 0 3 4 5** command must be defined for the **dial-shelf split backplane-ds0** *option* command to be active. You may also select the **user defined** option to define your own split.

Even if your system is already using a split dial shelf configuration, configuring one router shelf to handle two T3 trunks and the other router to handle the third trunk requires you to take the entire access server out of service. Busyout all connections before attempting to reconfigure. The configuration must be changed to setup one pool of TDM resources that can be used by either DMM cards or UPCs, and a second pool of two streams that contains TDM resources that can only be used by UPCs.

You may have more trunk capacity than 2048 calls. It is your decision how to provision the trunks so the backplane capacity is not exceeded. If more calls come in than backplane DS0 capacity for that half of the split, the call will be rejected and an error message printed for each call. This cannot be detected while a new configuration is being built because the router cannot tell which T1 trunks are provisioned and which are not. The user may want some trunks in hot standby.

The DMM, HMM, and VoIP cards can only use 1792 DS0 of the available 2048 backplane DS0. The UPC and trunk cards can use the full 2048 backplane DS0. The **show tdm splitbackplane** command will show the resources in two groups, the first 1792 accessible to all cards, and the remaining 256 accessible only to UPC and trunk cards.

For more information about split dial shelf configuration, refer to the *Cisco AS5800 Universal Access Server Split Dial Shelf Installation and Configuration Guide* and the hardware installation guides that accompanied your Cisco AS5800 Universal Access Server.

Configuring SPEs to Use an Upgraded Firmware File

To configure the SPEs to use the upgraded firmware file, use the following commands beginning in privileged EXEC mode to display the firmware version number:

	Command	Purpose
Step 1	`Router# show spe version`	Displays SPE firmware versions to obtain the On-Flash firmware filename.
Step 2	`Router# configure terminal`	Enters global configuration mode.

Step 3	AS5400:
	`Router(config)# spe slot/spe`
	or
	`Router(config)# spe slot/spe slot/spe`
	
	AS5800:
	`Router(config)# spe shelf/slot/spe`
	or
	`Router(config)# spe shelf/slot/spe shelf/slot/spe`
		Enters the SPE configuration mode. You can choose to configure a range of SPEs by specifying the first and last SPE in the range.

Step 4	`Router(config-spe)# firmware upgrade {busyout | download-maintenance | reboot}`	Specifies the upgrade method. Three methods of upgrade are available. The **busyout** keyword waits until all calls are terminated on an SPE before upgrading the SPE to the designated firmware. The **download-maintenance** keyword upgrades the firmware during the download maintenance time. The **reboot** keyword requests the access server to upgrade firmware at the next reboot.

	Command	Purpose
Step 5	`Router(config-spe)# firmware location filename`	Specifies the SPE firmware file in Flash memory to use for the selected SPEs. Allows you to upgrade firmware for SPEs after the new SPE firmware image is copied to your Flash memory. Enter the **no firmware location** command to revert back to the default Cisco IOS bundled SPE firmware.
Step 6	`Router(config-spe)# exit`	Exits SPE configuration mode.
Step 7	`Router# exit`	Exits global configuration mode.
Step 8	`Router# copy running-config startup-config`	Saves your changes.

Note The **copy ios-bundled** command is not necessary with UPCs or NextPort DFCs. By default, the version of SPE firmware bundled with the Cisco IOS software release transfers to all SPEs not specifically configured for a different SPE firmware file.

Disabling SPEs

To disable specific SPEs in the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use the following commands starting in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5400 Series Routers**
	`Router(config)# spe slot/spe`
	or
	`Router(config)# spe slot/spe slot/spe`
	Cisco AS5800 Series Routers
	`Router(config)# spe shelf/slot/spe`
	or
	`Router(config)# spe shelf/slot/spe shelf/slot/spe`
		Enters SPE configuration mode. You can also configure SPEs specifying the first and last SPE in a range.

	Command	Purpose
Step 2	`Router(config-spe)# busyout`	Gracefully disables an SPE by waiting for all the active services on the specified SPE to terminate. You can perform auto-diagnostic tests and firmware upgrades when you put the SPEs in the Busy out state. Active ports on the specified SPE will change the state of the specified range of SPEs to the BusyoutPending state. The state changes from BusyoutPending to Busiedout when all calls end. Use the **show spe** command to see the state of the range of SPEs. Use the **no** form of this command to re-enable the SPEs.
Step 3	`Router(config-spe)# shutdown`	Clears active calls on all ports on the SPE. Calls can no longer be placed on the SPE because the SPE state is changed to Busiedout. Use the **no** form of this command to re-enable the ports on the SPE.

Rebooting SPEs

To reboot specified SPEs, use the following command in privileged EXEC mode:

Command	Purpose

Cisco AS5400 Series Routers
`Router# clear spe slot/spe`
 	
Cisco AS5800 Series Routers
`Router# clear spe shelf/slot/spe`
	Allows manual recovery of a port that is frozen in a suspended state. Reboots SPEs that are in suspended or Bad state. Downloads configured firmware to the specified SPE or range of SPEs and power-on self test (POST) is executed.
	
	Note Depending on the problem, sometimes downloading the SPE firmware may not help recover a bad port or an SPE.
	
	This command can be executed regardless of the state of SPEs. All active ports running on the SPE are prematurely terminated, and messages are logged into the appropriate log.

Configuring Lines

To configure the lines to dial in to your network, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5400 Series Routers**
	`Router(config)# line slot/port slot/port`
	Cisco AS5800 Series Routers
	`Router(config)# line shelf/slot/port shelf/slot/port`
		Enters the line configuration mode. You can specify a range of slot and port numbers to configure.
		On the Cisco AS5400 access server, the NextPort DFC slot is defined as a value between 1 and 7. Slot 0 is reserved for the motherboard. Each NextPort DFC provides 18 SPEs. The SPE value ranges from 0 to 17. Since each SPE has six ports, the NextPort DFC has a total of 108 ports. The port value ranges from 0 to 107. To configure 108 ports on slot 3, you would enter **line 3/00 3/107**. If you wish to configure 324 ports on slots 3-5, you would enter **line 3/00 5/107**.
		On the Cisco AS5800 access server, the UPC slot is defined as a value between 2 and 11. Each UPC provides 54 SPEs. The SPE value ranges from 0 to 53. Because each SPE has six ports, the UPC has a total of 324 ports. The port value ranges from 0 to 323. To configure 324 ports on slot 3, you would enter **line 1/3/00 1/3/323**. If you want to configure 972 ports on slots 3-5, you would enter **line 1/3/00 1/5/323**.

Step 2	`Router(config-line)# transport input all`	Allows all protocols when connecting to the line.
Step 3	`Router(config-line)# autoselect ppp`	Enables remote IP users running a PPP application to dial in, bypass the EXEC facility, and connect directly to the network.
Step 4	`Router(config-line)# modem inout`	Enables incoming and outgoing calls.
Step 5	`Router(config-line)# modem autoconfigure type name`	Configures the attached modem using the entry for name.

Configuring Ports

This section describes how to configure Cisco AS5800 UPC or Cisco AS5400 NextPort DFC ports. You need to be in port configuration mode to configure these ports. The port configuration mode allows you to shut down or put individual ports or ranges of ports in busyout mode. To configure Cisco AS5800 UPC or Cisco AS5400 NextPort DFC ports, perform the following tasks beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5400 Series Routers**
	`Router(config)# port slot/spe`
	or
	`Router(config)# port slot/spe slot/spe`
	Cisco AS5800 Series Routers
	`Router(config)# port shelf/slot/spe`
	or
	`Router(config)# port shelf/slot/spe shelf/slot/spe`
		Enters port configuration mode. You can choose to configure a single port or range of ports.

Step 2	`Router(config-port)# busyout`	(Optional) Gracefully disables a port by waiting for the active services on the specified port to terminate. Use the **no** form of this command to re-enable the ports. Maintenance activities, such as testing, can still be performed while the port is in busyout mode. **Note** When a port is in busyout mode, the state of the SPE is changed to the consolidated states of all the underlying ports on that SPE.
Step 3	`Router(config-port)# shutdown`	(Optional) Clears active calls on the port. No more calls can be placed on the port in the shutdown mode. Use the **no** form of this command to re-enable the ports. **Note** When a port is in shutdown mode, the state of the SPE is changed to the consolidated states of all the underlying ports on that SPE.
Step 4	`Router(config-port)# exit`	Exits port configuration mode.

Verifying SPE Line and Port Configuration

To verify your SPE line configuration, enter the **show spe** command to display a summary for all the lines and ports:

Step 1 Enter the **show spe** command to display a summary for all the lines and ports:

```
Router# show spe
```

Step 2 Enter the **show line** command to display a summary for a single line.

AS5400

```
Router# show line 1/1
```

AS5800

```
Router# show line 1/2/10
```

Note If you are having trouble, make sure that you have turned on the protocols for If you are having trouble, make sure that you have turned on the protocols for connecting to the lines (**transport input all**) and that your access server is configured for incoming and outgoing calls (**modem inout**).

Configuring SPE Performance Statistics

Depending on the configuration, call record is displayed on the console, or the syslog, or on both. The log contains raw data in binary form, which must be viewed using the **show** commands listed in the section "Monitoring SPE Performance Statistics." You can configure some aspects of history events by using one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# spe call-record modem` *max-userid*	Requests the access server to generate a modem call record after a call is terminated. To disable this function, use the **no** form of this command.
`Router(config)# spe log-size` *number*	Sets the maximum size of the history event queue log entry for each port. The default is 50 events per port.

Clearing Log Events

To clear some or all of the log events relating to the SPEs as needed, use the following privileged EXEC mode commands:

Command	Purpose
`Router# clear spe log`	Clears all event entries in the slot history event log.
`Router# clear spe counters`	Clears statistical counters for all types of services for the specified SPE, a specified range of SPEs, or all SPEs. If you do not specify the range of SPEs or an SPE, the statistics for all SPEs are cleared.
`Router# clear port log`	Clears all event entries in the port level history event log. You cannot remove individual service events from the port log.

Troubleshooting SPEs

This section provides troubleshooting information for your SPEs regardless of service type mode.

Note SPE ports that pass the diagnostic test are marked as Pass, Fail, and Unkn. Ports that fail the diagnostic test are marked as Bad. These ports cannot be used for call connections. Depending on how many ports are installed, the diagnostic tests may take from 5 to 10 minutes to complete.

- Enter the **port modem startup-test** command to perform diagnostic testing for all modems during the system's initial startup or rebooting process. To disable the test, enter the **no port modem startup-test** command.
- Enter the **port modem autotest** command to perform diagnostic testing for all ports during the system's initial startup or rebooting process.To disable the test, enter the **no port modem autotest** command.

  You may additionally configure the following options:

  - Enter the **port modem autotest minimum** *ports* command to define the minimum number of free ports available for autotest to begin.
  - Enter the **port modem autotest time** *hh***:***mm interval* command to enable autotesting time and interval.
  - Enter the **port modem autotest error** *threshold* command to define the maximum number of errors detected for autotest to begin.
- Enter the **show port modem test** command to displays results of the SPE port startup test and SPE port auto-test.

When an SPE port is tested as Bad, you may perform additional testing by conducting a series of internal back-to-back connections and data transfers between two SPE ports. All port test connections occur inside the access server. For example, if mobile users cannot dial into port 2/5 (which is the sixth port on the NextPort DFC in the second chassis slot), attempt a back-to-back test with port 2/5 and a known-functioning port such as port 2/6.

- Enter the **test port modem back-to-back** *slot/port slot/port* command to perform internal back-to-back port tests between two ports sending test packets of the specified size.

Note You might need to enable this command on several different combinations of ports to determine which one is not functioning properly. A pair of operable ports successfully connects and completes transmitting data in both directions. An operable port and an inoperable port do not successfully connect with each other.

A sample back-to-back test might look like the following:

```
Router# test port modem back-to-back 2/10 3/20
Repetitions (of 10-byte packets) [1]:
*Mar  02 12:13:51.743:%PM_MODEM_MAINT-5-B2BCONNECT:Modems (2/10) and (3/20) connected
in back-to-back test:CONNECT33600/V34/LAP
*Mar  02 12:13:52.783:%PM_MODEM_MAINT-5-B2BMODEMS:Modems (3/20) and (2/10) completed
back-to-back test:success/packets = 2/2
```

Tips You may reboot the port that has problems using the **clear spe** EXEC command.

- Enter the **spe recovery** {**port-action** {**disable** | **recover** | **none**} | **port-threshold** *num-failures*} command to perform automatic recovery (removal from service and reloading of SPE firmware) of ports on an SPE at any available time.

  An SPE port failing to connect for a certain number of consecutive times indicates that a problem exists in a specific part or the whole of SPE firmware. Such SPEs have to be recovered by downloading firmware. Any port failing to connect *num-failures* times is moved to a state based on the **port-action** value, where you can choose to disable (mark the port as Bad) or recover the port when the SPE is in the idle state and has no active calls. The default for *num-failures* is 30 consecutive call failures.

Tips You may also schedule recovery using the **spe download maintenance** command.

- Enter the **spe download maintenance time** *hh*:*mm* | **stop-time** *hh*:*mm* | **max-spes** *number* | **window** *time-period* | **expired-window** {**drop-call** | **reschedule**} command to perform a scheduled recovery of SPEs.

  The download maintenance activity starts at the set start **time** and steps through all the SPEs that need recovery and the SPEs that need a firmware upgrade and starts maintenance on the maximum number of set SPEs for maintenance. The system waits for the **window** delay time for all the ports on the SPE to become inactive before moving the SPE to the Idle state. Immediately after the SPE moves to Idle state, the system starts to download firmware. If the ports are still in use by the end of **window** delay time, depending upon the **expired-window** setting, connections on the SPE ports are shutdown and the firmware is downloaded by choosing the **drop-call** option, or the firmware download is rescheduled to the next download maintenance time by choosing the **reschedule** option. This process continues until the number of SPEs under maintenance is below **max-spes,** or until **stop-time** (if set), or until all SPEs marked for recovery or upgrade have had their firmware reloaded.

Monitoring SPE Performance Statistics

This section documents various SPE performance statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs:

SPE Events and Firmware Statistics

To view SPE events and firmware statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use one or more of the following commands in privileged EXEC mode:

Command	Purpose

Cisco AS5400 series routers
`Router# show spe slot/spe`
Cisco AS5800 series routers
`Router# show spe shelf/slot/spe`
	Displays the SPE status for the specified range of SPEs.

`Router# show spe log [reverse | slot]`	Displays the SPE system log.
`Router# show spe version`	Lists all SPEs and the SPE firmware files used. **Note** This list helps you decide if you need to update your SPE firmware files.

Port Statistics

To view port statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use the following commands in privileged EXEC mode as needed:

Command	Purpose

Cisco AS5400 series routers
`Router# show port config {slot | slot/port}`
Cisco AS5800 series routers
`Router# show port config {slot | shelf/slot/port}`
	Displays the configuration information for specified ports or the specified port range. The port should have an active session associated at the time the command is executed.

Cisco AS5400 series routers `Router# show port digital log [reverse slot/port] [slot | slot/port]`	Displays the digital data event log.

Command	Purpose
Cisco AS5400 series routers `Router# show port modem log [reverse slot/port] [slot | slot/port]` **Cisco AS5800 series routers** Router# **show port modem log** [**reverse** *shelf/slot/port*] [*shelf/slot* | *shelf/slot/port*]	Displays the port history event log.

Cisco AS5400 series routers
`Router# show port modem test [slot | slot/port]`
Cisco AS5800 series routers
`Router# show port modem test [shelf/slot | shelf/slot/port]`
	Displays the test log for the specified SPE port range or all the SPE ports.

Cisco AS5400 series routers
`Router# show port operational-status [slot | slot/port]`
Cisco AS5800 series routers
`Router# show port operational-status [shelf/slot | shelf/slot/port]`
	Displays the operational status of the specified ports or the specified port range. The port should have an active session associated at the time the command is executed.

Digital SPE Statistics

To view digital SPE statistics for the Cisco AS5400 NextPort DFCs, use one or more of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show spe digital [slot | slot/spe]`	Displays history statistics of all digital SPEs.
`Router# show spe digital active [slot | slot/spe]`	Displays active digital statistics of a specified SPE, the specified range of SPEs, or all the SPEs.
`Router# show spe digital csr [summary | slot | slot/spe]`	Displays the digital call success rate statistics for a specific SPE, a range of SPEs, or all the SPEs.
`Router# show spe digital disconnect-reason [summary | slot | slot/spe]`	Displays the digital disconnect reasons for the specified SPE or range of SPEs. The disconnect reasons are displayed with Class boundaries.
`Router# show spe digital summary [slot | slot/spe]`	Displays digital history statistics of all SPEs, a specified SPE, or the specified range of SPEs for all service types.

SPE Modem Statistics

To view SPE modem statistics for the Cisco AS5400 NextPort DFCs or Cisco AS5800 UPCs, use one or more of the following commands in privileged EXEC mode:

Command	Purpose

Cisco AS5400 series routers
`Router# show spe modem active {slot | slot/spe}`
Cisco AS5800 series router:
`Router# show spe modem active {shelf/slot | shelf/slot/spe}`
	Displays the active statistics of a specified SPE, a specified range of SPEs, or all the SPEs serving modem traffic.

Cisco AS5400 series routers
`Router# show spe modem csr {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem csr {summary | shelf/slot | shelf/slot/spe}`
	Displays the call success rate statistics for a specific SPE, range of SPEs, or all the SPEs.

Cisco AS5400 series routers
`Router# show spe modem disconnect-reason {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem disconnect-reason {summary | shelf/slot | shelf/slot/spe}`
	Displays the disconnect reasons for the specified SPE or range of SPEs. The disconnect reasons are displayed with Class boundaries.

Cisco AS5400 series routers
`Router# show spe modem high speed {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem high speed {summary | shelf/slot | shelf/slot/spe}`
	Shows the connect-speeds negotiated within each high speed modulation or codecs for a specific range of SPEs or all the SPEs.

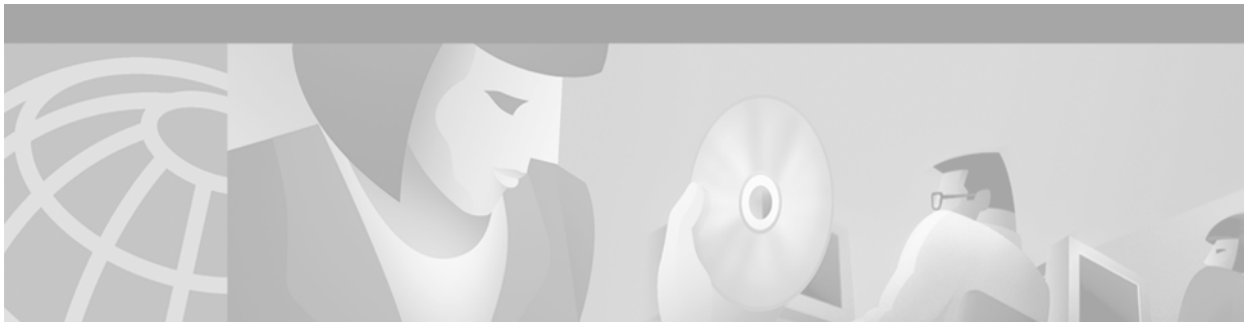
Cisco AS5400 series routers
`Router# show spe modem low speed {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem low speed {summary | shelf/slot | shelf/slot/spe}`
	Shows the connect-speeds negotiated within each low speed modulation or codecs for a specific range of SPEs or all the SPEs.

Cisco AS5400 series routers
`Router# show spe modem high standard {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem high standard {summary | shelf/slot | shelf/slot/spe}`
	Displays the total number of connections within each low modulation or codec for a specific range of SPEs.

Command	Purpose

Cisco AS5400 series routers
`Router# show spe modem low standard {summary | slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem low standard {summary | shelf/slot | shelf/slot/spe}`
	Displays the total number of connections within each high modulation or codec for a specific range of SPEs.

Cisco AS5400 series routers
`Router# show spe modem summary {slot | slot/spe}`
Cisco AS5800 series routers
`Router# show spe modem summary {shelf/slot | shelf/slot/spe}`
	Displays the history statistics of all SPEs, specified SPE or the specified range of SPEs.



Configuring and Managing External Modems

This chapter describes how to configure externally connected modems. These tasks are presented in the following main sections:

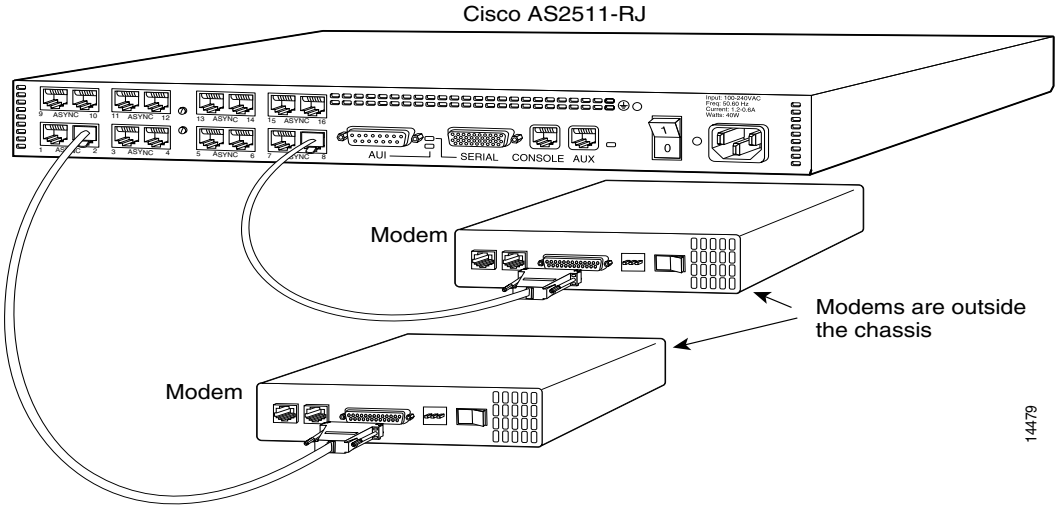
To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information, see the "Identifying Supported Platforms"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

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

External Modems on Low-End Access Servers

Some of the Cisco lower-end access servers, such as the Cisco AS2511-RJ shown in Figure 23, have cable connections to external modems. The asynchronous interfaces and lines are inside the access server.

Figure 23 Cisco AS2511-RJ Access Server



When you configure modems to function with your access server, you must provide initialization strings and other settings on the modem to tell it how to function with the access server.

This section assumes that you have already physically attached the modem to the access server. If not, refer to the user guide or installation and configuration guide for your access server for information about attaching modems.

Automatically Configuring an External Modem

The Cisco IOS software can issue initialization strings automatically, in a file called a modemcap, for most types of modems externally attached to the access server. A modemcap is a series of parameter settings that are sent to your modem to configure it to interact with the Cisco device in a specified way. The Cisco IOS software defines modemcaps that have been found to properly initialize most modems so that they function properly with Cisco routers and access servers. For Cisco IOS Release 12.2, these modemcaps have the following names:

- default—Generic Hayes interface external modem
- codex_3260—Motorola Codex 3260 external
- usr_courier—U.S. Robotics Courier external
- usr_sportster—U.S. Robotics Sportster external
- hayes_optima—Hayes Optima external[1]
- global_village—Global Village Teleport external
- viva—Viva (Rockwell ACF with MNP) external
- telebit_t3000—Telebit T3000 external
- nec_v34—NEC V.34 external
- nec_v110—NEC V.110 TA external
- nec_piafs—NEC PIAFS TA external

[1]The hayes_optima modemcap is not recommended for use; instead, use the default modemcap.

Enter these modemcap names with the **modemcap entry** command.

If your modem is not on this list and if you know what modem initialization string you need to use with it, you can create your own modemcap; see the following procedure "Using the Modem Autoconfigure Type Modemcap Feature." To have the Cisco IOS software determine what type of modem you have, use the **modem autoconfigure discovery** command to configure it, as described in the procedure "Using the Modem Autoconfigure Discovery Feature."

Using the Modem Autoconfigure Type Modemcap Feature

Step 1 Use the **modemcap edit** command to define your own modemcap entry.

The following example defines modemcap MODEMCAPNAME:

```
Router(config)# modemcap edit MODEMCAPNAME miscellaneous &FS0=1&D3
```

Step 2 Apply the modemcap to the modem lines as shown in the following example:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# line 33 34
Router(config-line)# modem autoconfigure type MODEMCAPNAME
Router(config-line)#
Jan 16 18:12:59.643: TTY34: detection speed (115200) response ---OK---
Jan 16 18:12:59.643: TTY34: Modem command:  --AT&FS0=1&D3--
Jan 16 18:12:59.659: TTY33: detection speed (115200) response ---OK---
Jan 16 18:12:59.659: TTY33: Modem command:  --AT&FS0=1&D3--
Jan 16 18:13:00.227: TTY34: Modem configuration succeeded
Jan 16 18:13:00.227: TTY34: Detected modem speed 115200
Jan 16 18:13:00.227: TTY34: Done with modem configuration
Jan 16 18:13:00.259: TTY33: Modem configuration succeeded
Jan 16 18:13:00.259: TTY33: Detected modem speed 115200
Jan 16 18:13:00.259: TTY33: Done with modem configuration
```

Using the Modem Autoconfigure Discovery Feature

If you prefer the modem software to use its autoconfigure mechanism to configure the modem, use the **modem autoconfigure discovery** command.

The following example shows how to configure modem autoconfigure discovery mode:

```
Router# terminal monitor
Router# debug confmodem
Modem Configuration Database debugging is on
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# line 33 34
Router(config-line)# modem autoconfigure discovery
Jan 16 18:16:17.724: TTY33: detection speed (115200) response ---OK---
Jan 16 18:16:17.724: TTY33: Modem type is default
Jan 16 18:16:17.724: TTY33: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:17.728: TTY34: detection speed (115200) response ---OK---
Jan 16 18:16:17.728: TTY34: Modem type is default
Jan 16 18:16:17.728: TTY34: Modem command:  --AT&F&C1&D2S0=1H0--
Jan 16 18:16:18.324: TTY33: Modem configuration succeeded
```

```
Jan 16 18:16:18.324: TTY33: Detected modem speed 115200
Jan 16 18:16:18.324: TTY33: Done with modem configuration
Jan 16 18:16:18.324: TTY34: Modem configuration succeeded
Jan 16 18:16:18.324: TTY34: Detected modem speed 115200
Jan 16 18:16:18.324: TTY34: Done with modem configuration
```

Manually Configuring an External Modem

If you cannot configure your modem automatically, you must configure it manually. This section describes how to determine and issue the correct initialization string for your modem and how to configure your modem with it.

Modem command sets vary widely. Although most modems use the Hayes command set (prefixing commands with **at**), Hayes-compatible modems do not use identical **at** command sets.

Refer to the documentation that came with your modem to learn how to examine the current and stored configuration of the modem that you are using. Generally, you enter **at** commands such as **&v**, **i4**, or ***o** to view, inspect, or observe the settings.

Timesaver

You must first create a direct Telnet or connection session to the modem before you can send an initialization string. You can use **AT&F** as a basic modem initialization string in most cases. To establish a direct Telnet session to an external modem, determine the IP address of your LAN (Ethernet) interface, and then enter a Telnet command to port 2000 + *n* on the access server, where *n* is the line number to which the modem is connected. See the sections "Testing the Modem Connection" and "Managing Telnet Sessions" for more information about making Telnet connections.

A sample modem initialization string for a US Robotics Courier modem is as follows:

```
&b1&h1&r2&c1&d3&m4&k1s0=1
```

Modem initialization strings enable the following functions:

- Locks the speed of the modem to the speed of the serial port on the access server
- Sets hardware flow control (RTS/CTS or request to send/clear to send)
- Ensures correct data carrier detect (DCD) operation
- Ensures proper data terminal ready (DTR) interpretation
- Answers calls on the first ring

Note

Make sure to turn off automatic baud rate detection because the modem speeds must be set to a fixed value.

The port speed must not change when a session is negotiated with a remote modem. If the speed of the port on the access server is changed, you must establish a direct Telnet session to the modem and send an **at** command so that the modem can learn the new speed.

Modems differ in the method that they use to lock the EIA/TIA-232 (serial) port speed. In the modem documentation, vendors use terms such as port-rate adjust, speed conversion, or buffered mode. Enabling error correction often puts the modem in the buffered mode. Refer to your modem documentation to learn how your modem locks speed (check the settings **&b**, **\j**, **&q**, **\n**, or s-register settings).

RTS and CTS signals must be used between the modem and the access server to control the flow of data. Incorrectly configuring flow control for software or setting no flow control can result in hung sessions and loss of data. Modems differ in the method that they use to enable hardware flow control. Refer to your modem documentation to learn how to enable hardware flow control (check the settings **&e**, **&k**, **&h**, **&r**, or s-register).

The modem must use the DCD wire to indicate to the access server when a session has been negotiated and is established with a remote modem. Most modems use the setting **&c1**. Refer to your modem documentation for the DCD settings used with your modem.

The modem must interpret a toggle of the DTR signal as a command to drop any active call and return to the stored settings. Most modems use the settings **&d2** or **&d3**. Refer to your modem documentation for the DTR settings used with your modem.

If a modem is used to service incoming calls, it must be configured to answer a call after a specific number of rings. Most modems use the setting **s0=1** to answer the call after one ring. Refer to your modem documentation for the settings used with your modem.

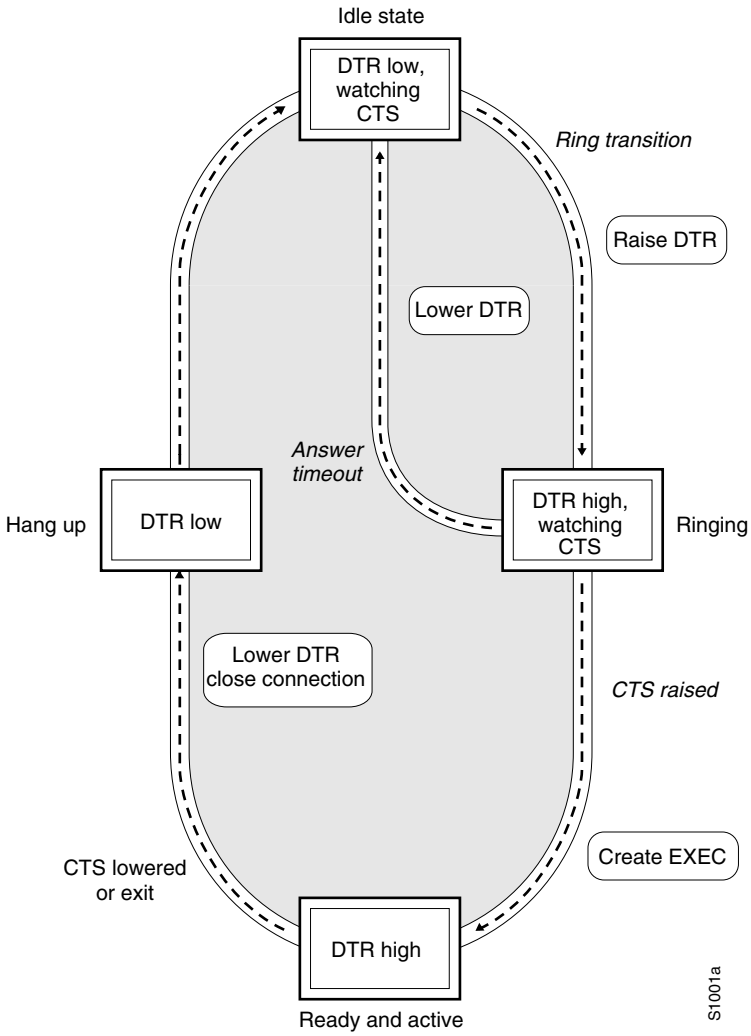
Supporting Dial-In Modems

The Cisco IOS software supports dial-in modems that use DTR to control the off-hook status of the telephone line. This feature is supported primarily on old-style modems, especially those in Europe. To configure the line to support this feature, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem callin`	Configures a line for a dial-in modem.

Figure 24 illustrates the **modem callin** command. When a modem dialing line is idle, it has its DTR signal at a low state and waits for a transition to occur on the data set ready (DSR) input. This transition causes the line to raise the DTR signal and start watching the CTS signal from the modem. After the modem raises CTS, the Cisco IOS software creates an EXEC session on the line. If the timeout interval (set with the **modem answer-timeout** command) passes before the modem raises the CTS signal, the line lowers the DTR signal and returns to the idle state.

Figure 24 EXEC Creation on a Line Configured for Modem Dial-In



Note The **modem callin** and **modem cts-required** line configuration commands are useful for SLIP operation. These commands ensure that when the line is hung up or the CTS signal drops, the line reverts from Serial Line Internet Protocol (SLIP) mode to normal interactive mode. These commands do not work if you put the line in network mode permanently.

Although you can use the **modem callin** line configuration command with newer modems, the **modem dialin** line configuration command described in this section is more appropriate. The **modem dialin** command frees up CTS input for hardware flow control. Modern modems do not require the assertion of DTR to answer a phone line (that is, to take the line off-hook).

Testing the Modem Connection

To test the connection, send the modem the AT command to request its attention. The modem should respond with "OK." For example:

```
at
OK
```

If the modem does not reply to the **at** command, perform the following steps:

Step 1 Enter the **show users** EXEC command and scan the display output. The output should not indicate that the line is in use. Also verify that the line is configured for **modem inout**.

Step 2 Enter the **show line** EXEC command. The output should contain the following two lines:

```
Modem state: Idle
Modem hardware state: CTS noDSR  DTR RTS
```

If the output displays "no CTS" for the modem hardware state, the modem is not connected, is not powered up, is waiting for data, or might not be configured for hardware flow control.

Step 3 Verify the line speed and modem transmission rate. Make sure that the line speed on the access server matches the transmission rate, as shown in Table 13.

Table 13 Matching Line Speed with Transmission Rate

Modem Transmission Rate (in bits per second)	Line Speed on the Access Server (in bits per second)
9600	38400
14400	57600
28800	115200

To verify the line speed, use the **show run** EXEC command. The line configuration fragment appears at the tail end of the output.

The following example shows that lines 7 through 9 are transmitting at 115200 bits per second (bps). Sixteen 28800-kbps modems are connected to a Cisco AS2511-RJ access server via a modem cable.

```
Router# show run

Building configuration...

Current configuration:
. . .
!
 line 1 16
  login local
  modem InOut
  speed 115200
  transport input all
  flowcontrol hardware
  script callback callback
  autoselect ppp
  autoselect during-login
```

Step 4 The speeds of the modem and the access server are likely to be different. If so, switch off the modem, and then switch it back on. This action should change the speed of the modem to match the speed of the access server.

Step 5 Check your cabling and the modem configuration (echo or result codes might be off). Enter the appropriate **at** modem command to view the modem configuration, or use the **at&f** command to return to factory defaults. Refer to your modem documentation to learn the appropriate **at** command to view your modem configuration.

Note See the section "Configuring Cisco Integrated Modems Using Modem Attention Commands" in the "Configuring and Managing Integrated Modems" chapter for information about modem attention commands for the Cisco internal modems.

Managing Telnet Sessions

You communicate with an external modem by establishing a direct Telnet session from the asynchronous line on the access server, which is connected to the modem. This process is also referred to as *reverse Telnet*. Performing a reverse Telnet means that you are initiating a Telnet session out the asynchronous line, instead of accepting a connection into the line (called a *forward* connection).

Note Before attempting to allow inbound connections, make sure that you close all open connections to the modems attached to the access server. If you have a modem port in use, the modem will not accept a call properly.

To establish a direct Telnet session to an external modem, determine the IP address of your LAN (Ethernet) interface, and then enter a Telnet command to port 2000 + *n* on the access server, where *n* is the line number to which the modem is connected. For example, to connect to the modem attached to line 1, enter the following command from an EXEC session on the access server:

```
Router# telnet 172.16.1.10 2001
Trying 172.16.1.10, 2001 ... Open
```

This example enables you to communicate with the modem on line 1 using the AT (attention) command set defined by the modem vendor.

Timesaver Use the **ip host** configuration command to simplify direct Telnet sessions with modems. The **ip host** command maps an IP address of a port to a device name. For example, the **modem1 2001 172.16.1.10** command enables you to enter **modem1** to initiate a connection with the modem, instead of repeatedly entering **telnet 172.16.1.10 2001** each time you want to communicate with the modem.

You can also configure asynchronous rotary line queueing, which places Telnet login requests in a queue when lines are busy. See the section "Configuring Asynchronous Rotary Line Queueing" in the "Configuring Asynchronous Lines and Interfaces" chapter for more information.

Suspending Telnet Sessions:

When you are connected to an external modem, the direct Telnet session must be terminated before the line can accept incoming calls. If you do not terminate the session, it will be indicated in the output of the **show users** command and will return a modem state of ready if the line is still in use. If the line is no longer in use, the output of the **show line** *value* command will return a state of idle. Terminating the Telnet session requires first suspending it, then disconnecting it.

To suspend a Telnet session, perform the following steps:

Step 1 Enter Ctrl-Shift-6 x to suspend the Telnet session:

```
- suspend keystroke -
Router#
```

Note Ensure that you can reliably issue the escape sequence to suspend a Telnet session. Some terminal emulation packages have difficulty sending the Ctrl-Shift-6 x sequence. Refer to your terminal emulation documentation for more information about escape sequences.

Step 2 Enter the **where** EXEC command to check the connection numbers of open sessions:

```
Router# where
Conn Host                Address             Byte  Idle Conn Name
*  1 172.16.1.10         172.16.1.10            0     0  172.16.1.10
   2 172.16.1.11         172.16.1.11            0    12  modem2
```

Step 3 When you have suspended a session with one modem, you can connect to another modem and suspend it:

```
Router# telnet modem2
Trying modem2 (172.16.1.11, 2002) ... Open

- suspend keystroke -
Router#
```

Step 4 To disconnect (completely close) a Telnet session, enter the **disconnect** EXEC command:

```
Router# disconnect line 1
Closing connection to 172.16.1.10 [confirm] y
Router# disconnect line 2
Closing connection to 172.16.1.11 [confirm] y
Router#
```

Modem Troubleshooting Tips

Table 14 contains troubleshooting tips on modem access and control.

Table 14 Modem Troubleshooting Tips

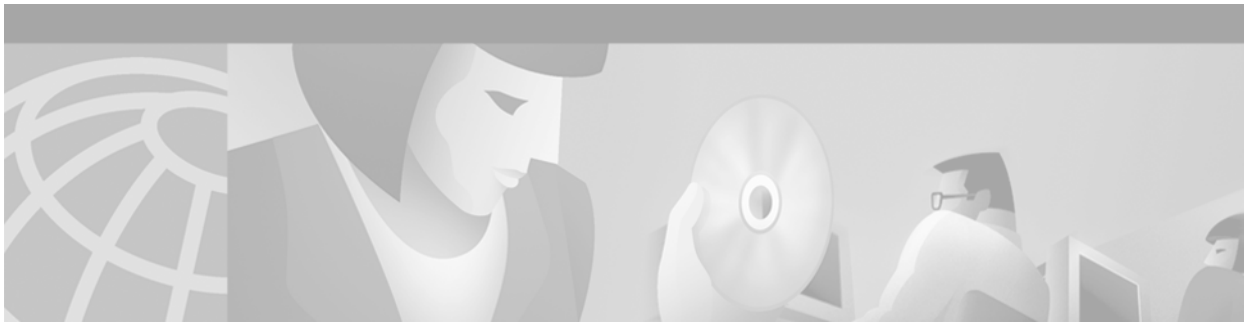
Problem	Likely Cause
Connection refused.	Someone already has a connection to that port. or an EXEC is running on that port. or The modem failed to lower the carrier detect (CD) signal after a call disconnected, resulting in an EXEC that remained active after disconnect. To force the line back into an idle state, clear the line from the console and try again. If it still fails, ensure that you have set **modem inout** command for that line. If you don't have modem control, either turn off EXEC on the line (by using the **exec-timeout** line configuration command) before making a reverse connection or configure the modem using an external terminal. As a last resort, disconnect the modem, clear the line, make the Telnet connection, and then attach the modem. The prevents a misconfigured modem from denying you line access.
Connection appears to hang.	Try entering "^U" (clear line), "^Q" (XON), and press Return a few times to try to establish terminal control.
EXEC does not come up; autoselect is on.	Press Return to enter EXEC.
Modem does not hang up after entering **quit**.	The modem is not receiving DTR information, or you have not set up modem control on the router.
Interrupts another user session when you dial in.	The modem is not dropping CD on disconnect, or you have not set up modem control on the router.
Connection hangs after entering "+++" on the dialing modem, followed by an ATO.	The answering modem saw and interpreted the "+++" when it was echoed to you. This is a bug in the answering modem, common to many modems. There may be a switch to work around this problem; check the modem's documentation.
Losing data.	You may have Hardware Flow Control only on for either the router's line (DTE) or the modem (DCE). Hardware Flow Control should be on for both or off for both, but not for only one.
Using MDCE.	Turn MDCE into an MMOD by moving pin 6 to pin 8 because most modems use CD and not DSR to indicate the presence of carrier. You can also program some modems to provide carrier info via DSR.

Checking Other Modem Settings

This section defines other settings that might be needed or desirable, depending on your modem.

Error correction can be negotiated between two modems to ensure a reliable data link. Error correction standards include Link Access Procedure for Modems (LAPM) and MNP4. V.42 error correction allows either LAPM or MNP4 error correction to be negotiated. Modems differ in the way they enable error correction. Refer to your modem documentation for the error correction methods used with your modem.

Data compression can be negotiated between two modems to allow for greater data throughput. Data compression standards include V.42*bis* and MNP5. Modems differ in the way they enable data compression. Refer to your modem documentation for the data compression settings used with your modem.



Modem Signal and Line States

This chapter describes modem states in the following section:

- Signal and Line State Diagrams

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Modem Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Signal and Line State Diagrams

The following signal and line state diagrams accompany some of the tasks in the following sections to illustrate how the modem control works:

- Configuring Automatic Dialing
- Automatically Answering a Modem
- Supporting Dial-In and Dial-Out Connections
- Configuring a Line Timeout Interval
- Closing Modem Connections
- Configuring a Line to Disconnect Automatically
- Supporting Reverse Modem Connections and Preventing Incoming Calls

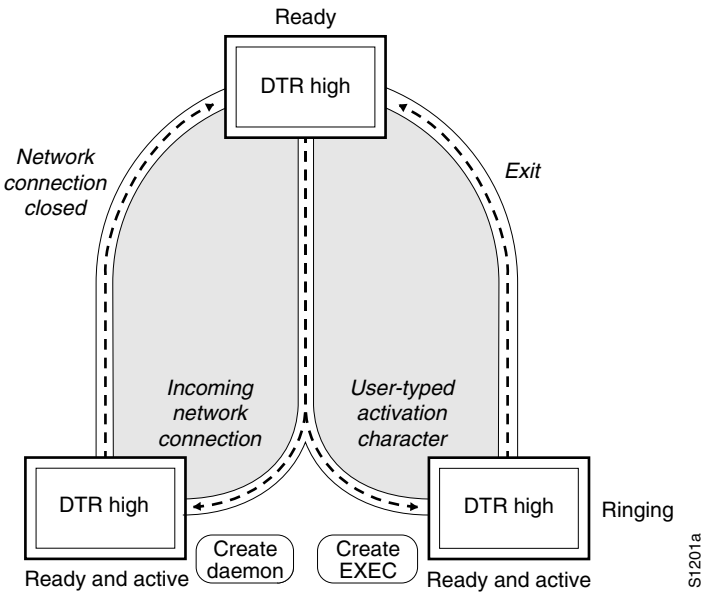
The diagrams show two processes:

- The “create daemon” process creates a tty daemon that handles the incoming network connection.
- The “create EXEC” process creates the process that interprets user commands. (See Figure 25 through Figure 29.)

In the diagrams, the current signal state and the signal the line is watching are listed inside each box. The state of the line (as displayed by the **show line** EXEC command) is listed next to the box. Events that change that state appear in italics along the event path, and actions that the software performs are described within ovals.

Figure 25 illustrates line states when no modem control is set. The DTR output is always high, and CTS and RING are completely ignored. The Cisco IOS software starts an EXEC session when the user types the activation character. Incoming TCP connections occur instantly if the line is not in use and can be closed only by the remote host.

Figure 25 EXEC and Daemon Creation on a Line with No Modem Control



Configuring Automatic Dialing

With the dialup capability, you can set a modem to dial the phone number of a remote router automatically. This feature offers cost savings because phone line connections are made only when they are needed—you pay for using the phone line only when there is data to be received or sent.

To configure a line for automatic dialing, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem dtr-active`	Configures a line to initiate automatic dialing.

Using the **modem dtr-active** command causes a line to raise DTR signal only when there is an outgoing connection (such as reverse Telnet, NetWare Asynchronous Support Interface (NASI), or DDR), rather than leave DTR raised all the time. When raised, DTR potentially tells the modem that the router is ready to accept a call.

Automatically Answering a Modem

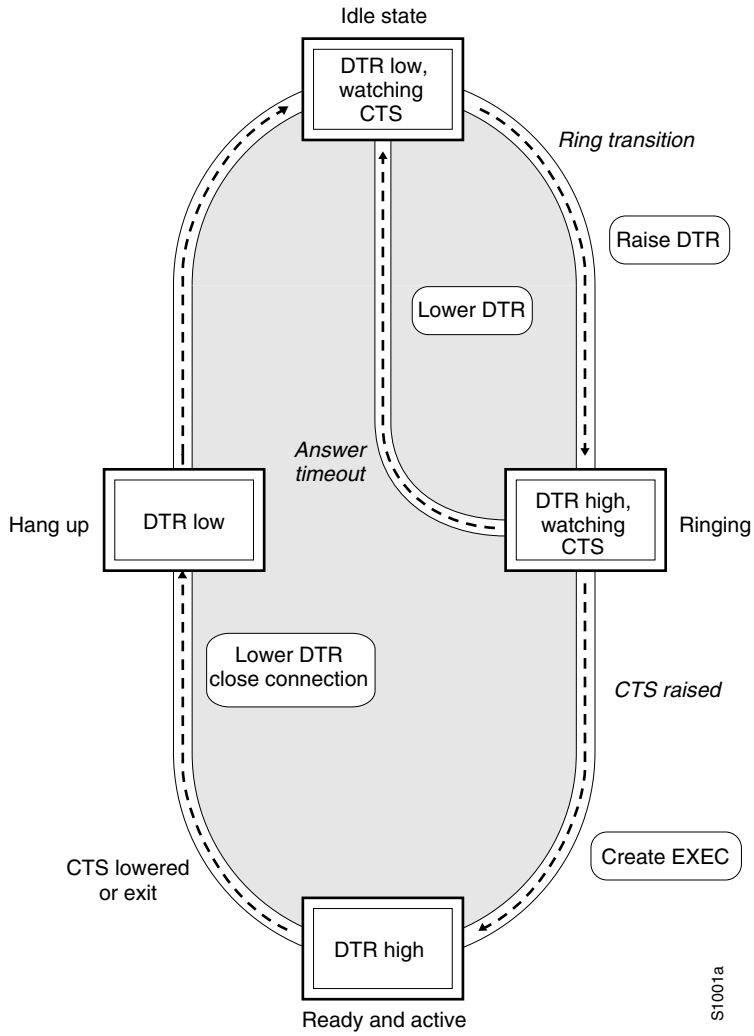
You can configure a line to answer a modem automatically. You also can configure the modem to answer the telephone on its own (as long as DTR is high), drop connections when DTR is low, and use its Carrier Detect (CD) signal to accurately reflect the presence of carrier. (Configuring the modem is a modem-dependent process.) First, wire the modem CD signal (generally pin-8) to the router RING input (pin-22), then use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem dialin`	Configures a line to automatically answer a modem.

You can turn on modem hardware flow control independently to respond to the status of router CTS input. Wire CTS to whatever signal the modem uses for hardware flow control. If the modem expects to control hardware flow in both directions, you might also need to wire modem flow control input to some other signal that the router always has high, such as the DTR signal.

Figure 26 illustrates the **modem dialin** process with a high-speed dialup modem. When the Cisco IOS software detects a signal on the RING input of an idle line, it starts an EXEC or autobaud process on that line. If the RING signal disappears on an active line, the Cisco IOS software closes any open network connections and terminates the EXEC facility. If the user exits the EXEC or the software terminates because of no user input, the line makes the modem hang up by lowering the DTR signal for 5 seconds. After 5 seconds, the modem is ready to accept another call.

Figure 26 EXEC Creation on a Line Configured for a High-Speed Modem



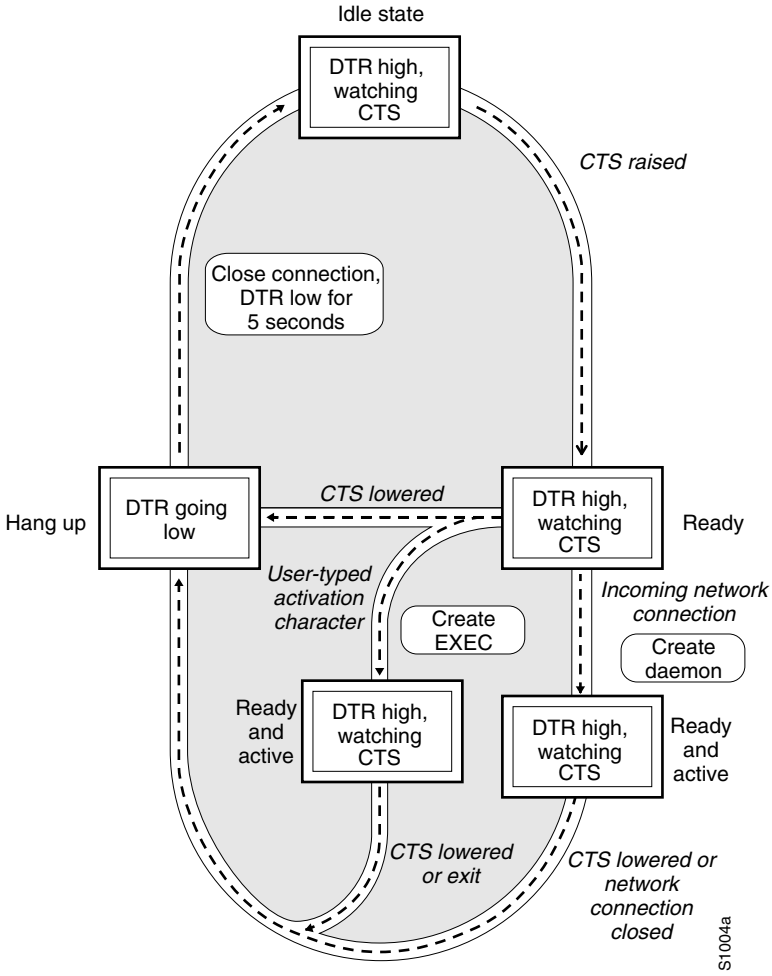
Supporting Dial-In and Dial-Out Connections

To configure a line for both incoming and outgoing calls, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem inout`	Configures a line for both incoming and outgoing calls.

Figure 27 illustrates the **modem inout** command. If the line is activated by raising the data set ready (DSR) signal, it functions exactly as a line configured with the **modem dialin** line configuration command described in the section "Automatically Answering a Modem" earlier in this chapter. If the line is activated by an incoming TCP connection, the line functions similarly to lines not used with modems.

Figure 27 EXEC and Daemon Creation for Incoming and Outgoing Calls



Note If your system incorporates dial-out modems, consider using access lists to prevent unauthorized use.

Configuring a Line Timeout Interval

To change the interval that the Cisco IOS software waits for the CTS signal after raising the DTR signal in response to the DSR (the default is 15 seconds), use the following command in line configuration mode. The timeout applies to the **modem callin** command only.

Command	Purpose
`Router(config-line)# modem answer-timeout seconds`	Configures modem line timing.

Note The DSR signal is called RING on older ASM-style chassis.

Closing Modem Connections

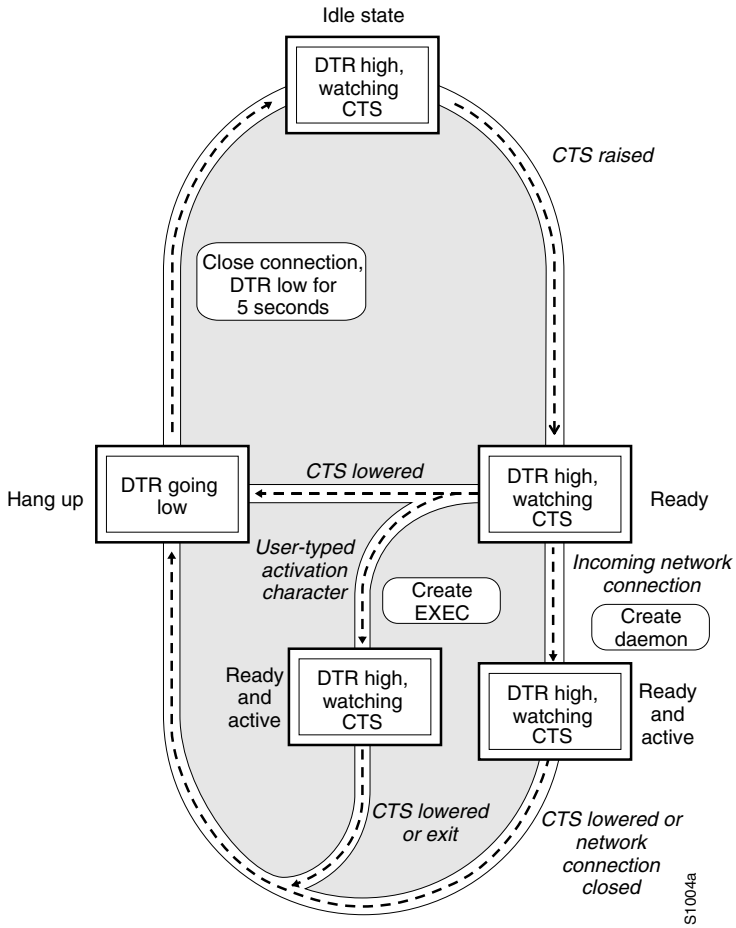
Note The **modem cts-required** command was replaced by the **modem printer** command in Cisco IOS Release 12.2.

To configure a line to close connections from a user's terminal when the terminal is turned off and to prevent inbound connections to devices that are out of service, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem cts-required`	Configures a line to close connections.

Figure 28 illustrates the **modem cts-required** command operating in the context of a continuous CTS signal. This form of modem control requires that the CTS signal be high for the entire session. If CTS is not high, the user input is ignored and incoming connections are refused (or sent to the next line in a rotary group).

Figure 28 EXEC and Daemon Creation on a Line Configured for Continuous CTS



Configuring a Line to Disconnect Automatically

To configure automatic line disconnect, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autohangup`	Configures automatic line disconnect.

The **autohangup** command causes the EXEC facility to issue the **exit** command when the last connection closes. This feature is useful for UNIX-to-UNIX copy program (UUCP) applications because UUCP scripts cannot issue a command to hang up the telephone. This feature is not used often.

Supporting Reverse Modem Connections and Preventing Incoming Calls

In addition to initiating connections, the Cisco IOS software can receive incoming connections. This capability allows you to attach serial and parallel printers, modems, and other shared peripherals to the router or access server and drive them remotely from other modem-connected systems. The Cisco IOS software supports reverse TCP, XRemote, and local-area transport (LAT) connections.

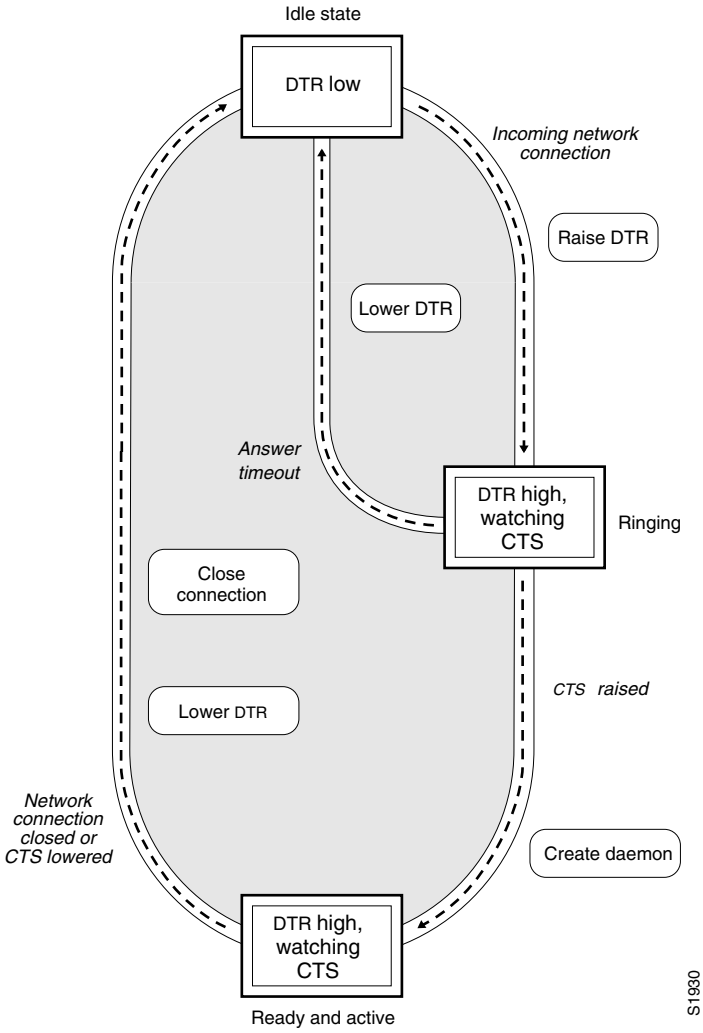
The specific TCP port or socket to which you attach the device determines the type of service that the Cisco IOS software provides on a line. When you attach the serial lines of a computer system or a data terminal switch to the serial lines of the access server, the access server can act as a network front-end device for a host that does not support the TCP/IP protocols. This arrangement is sometimes called *front-ending* or *reverse connection mode*.

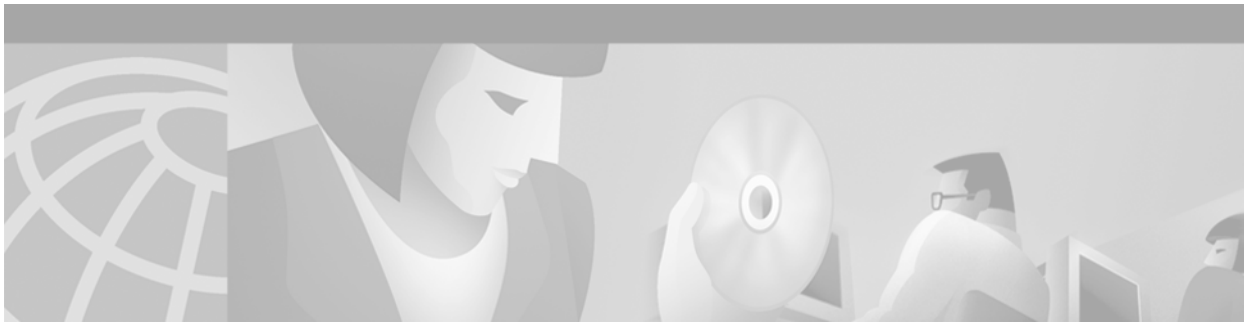
The Cisco IOS software supports ports connected to computers that are connected to modems. To configure the Cisco IOS software to function somewhat like a modem, use the following command in line configuration mode. This command also prevents incoming calls.

Command	Purpose
`Router(config-line)# modem callout`	Configures a line for reverse connections and prevents incoming calls.

Figure 29 illustrates the **modem callout** process. When the Cisco IOS software receives an incoming connection, it raises the DTR signal and waits to see if the CTS signal is raised to indicate that the host has noticed the router DTR signal. If the host does not respond within the interval set by the **modem answer-timeout** line configuration command, the software lowers the DTR signal and drops the connection.

Figure 29 Daemon Creation on a Line Configured for Modem Dial-Out





Creating and Using Modem Chat Scripts

This chapter describes how to create and use modem chat scripts. These tasks are presented in the following main sections:

- Chat Script Overview
- How To Configure Chat Scripts
- Using Chat Scripts

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the modem support commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference* publication. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Chat Script Overview

Chat scripts are strings of text used to send commands for modem dialing, logging in to remote systems, and initializing asynchronous devices connected to an asynchronous line.

Note On a router, chat scripts can be configured only on the auxiliary port.

A chat script must be configured to dial out on asynchronous lines. You also can configure chat scripts so that they can be executed automatically for other specific events on a line, or so that they are executed manually.

Each chat script is defined for a different event. These events can include the following:

- Line activation
- Incoming connection initiation
- Asynchronous dial-on-demand routing (DDR)
- Line resets
- Startup

Note Outbound chat scripts are not supported on lines where modem control is set for inbound activity only using the **modem dialin** command.

How To Configure Chat Scripts

The following tasks must be performed before a chat script can be used:

- Define the chat script in global configuration mode using the **chat-script** command.
- Configure the line so that a chat script is activated when a specific event occurs (using the **script** line configuration command), or start a chat script manually (using the **start-chat** privileged EXEC command).

To configure a chat script, perform the tasks in the following sections:

- Understanding Chat Script Naming Conventions (Required)
- Creating a Chat Script (Required)
- Configuring the Line to Activate Chat Scripts (Required)
- Manually Testing a Chat Script on an Asynchronous Line (Optional)

See the section "Using Chat Scripts" later in this chapter for examples of how to use chat scripts.

Understanding Chat Script Naming Conventions

When you create a script name, include the modem vendor, type, and modulation, separated by hyphens, as follows:

vendor-type-modulation

For example, if you have a Telebit t3000 modem that uses V.32*bis* modulation, your script name would be*:*

```
telebit-t3000-v32bis
```

Note Adhering to the recommended naming convention allows you to specify a range of chat scripts by using partial names in UNIX-style regular expressions. The regular expressions are used to match patterns and select chat scripts to use. This method is particularly useful for dialer rotary groups on an interface that dials multiple destinations. Regular expressions are described in the "Regular Expressions" appendix in the *Cisco IOS Terminal Services Configuration Guide*.

Creating a Chat Script

We recommend that one chat script (a "modem" chat script) be written for placing a call and that another chat script (a "system" or "login" chat script) be written to log in to remote systems, where required.

To define a chat script, use the following command in global configuration mode:

Command	Purpose
`Router(config)# chat-script script-name expect send...`	Creates a script that will place a call on a modem, log in to a remote system, or initialize an asynchronous device on a line.

The Cisco IOS software waits for the string from the modem (defined by the *expect* portion of the script) and uses it to determine what to send back to the modem (defined by the *send* portion of the script).

Chat String Escape Key Sequences

Chat script send strings can include the special escape sequences listed in Table 15.

Table 15 Chat Script Send String Escape Sequences

Escape Sequence	Description
\	Sends the ASCII character with its octal value.
\\	Sends a backslash (\) character.
\”	Sends a double-quote (“) character (does not work *within* double quotes).
\c	Suppresses a new line at the end of the send string.
\d	Delays for 2 seconds.
\K	Inserts a BREAK.
\n	Sends a newline or linefeed character.
\N	Sends a null character.
\p	Pauses for 0.25 second.
\q	Reserved, not yet used.
\r	Sends a return.
\s	Sends a space character.
\t	Sends a tab character.
\T	Replaced by phone number.
“ ”	Expects a null string.
BREAK	Causes a BREAK. This sequence is sometimes simulated with line speed changes and null characters. May not work on all systems.
EOT	Sends an end-of-transmission character.

Adding a Return Key Sequence

After the connection is established and you press the Return key, you must often press Return a second time before the prompt appears. To create a chat script that enters this additional Return key for you, include the following string with the Return key escape sequence (see Table 15) as part of your chat script:

```
ssword:-/r-ssword
```

This part of the script specifies that, after the connection is established, you want **ssword** to be displayed. If it is not displayed, you must press Return again after the timeout passes. (For more information about expressing characters in chat scripts, see the "Regular Expressions" appendix in the *Cisco IOS Terminal Services Configuration Guide*.)

Chat String Special-Case Script Modifiers

Special-case script modifiers are also supported; refer to Table 16 for examples.

Table 16 Special-Case Script Modifiers

Special Case	Function
ABORT *string*	Designates a string whose presence in the input indicates that the chat script has failed. (You can have as many active abort entries as you like.)
TIMEOUT *time*	Sets the time to wait for input, in seconds. The default is 5 seconds, and a timeout of 60 seconds is recommended for V.90 modems.

For example, if a modem reports BUSY when the number dialed is busy, you can indicate that you want the attempt stopped at this point by including ABORT BUSY in your chat script.

Note If you use the *expect-send* pair ABORT SINK instead of ABORT ERROR, the system terminates abnormally when it encounters SINK instead of ERROR.

Configuring the Line to Activate Chat Scripts

Chat scripts can be activated by any of five events, each corresponding to a different version of the **script** line configuration command. To start a chat script manually at any point, see the following section, "Manually Testing a Chat Script on an Asynchronous Line."

To define a chat script to start automatically when a specific event occurs, use one of the following commands in line configuration mode:

Command	Purpose
`Router(config-line)# script activation` *regexp*[1]	Starts a chat script on a line when the line is activated (every time a command EXEC is started on the line).
`Router(config-line)# script connection` *regexp*	Starts a chat script on a line when a network connection is made to the line.
`Router(config-line)# script dialer` *regexp*	Specifies a modem script for DDR on a line.
`Router(config-line)# script reset` *regexp*[2]	Starts a chat script on a line whenever the line is reset.
`Router(config-line)# script startup` *regexp*[2]	Starts a chat script on a line whenever the system is started up.

1. The *regexp* argument is a regular expression that is matched to a script name that has already been defined using the **chat-script** command.
2. Do not use the **script reset** or **script startup** commands to configure a modem; instead use the **modem autoconfigure** command.

Note Outbound chat scripts are not supported on lines where modem control is set for inbound activity only (using the **modem dialin** command).

Manually Testing a Chat Script on an Asynchronous Line

To test a chat script on any line that is currently not active, use the following commands in privileged EXEC mode:

	Command	Purpose
Step 1	`Router# **debug chat line** *number*`	Starts detailed debugging on the specified line.
Step 2	`Router# **start-chat** *regexp* [*line-number* [*dialer-string*]]`	Starts a chat script on any asynchronous line.

If you do not specify the line number, the script runs on the current line. If the line specified is already in use, you cannot start the chat script. A message appears indicating that the line is already in use.

Using Chat Scripts

The following sections provide examples of how to use chat scripts:

- Generic Chat Script Example
- Traffic-Handling Chat Script Example
- Modem-Specific Chat Script Examples
- Dialer Mapping Example
- System Login Scripts and Modem Script Examples

Generic Chat Script Example

The following example chat script includes a pair of empty quotation marks (“ ”), which means “expect anything,” and \r, which means “send a return”:

```
" " \r "name:" "myname" "ord":" "mypassword" ">" "slip default"
```

Traffic-Handling Chat Script Example

The following example shows a configuration in which, when there is traffic, a random line will be used. The dialer code will try to find a script that matches either the modem script *.*-v32* or the system script cisco. If there is no match for either the modem script or the system script, you will see a “no matching chat script found” message.

```
interface dialer 1
! v.32 rotaries are in rotary 1.
dialer rotary-group 1
! Use v.32 generic script.
dialer map ip 10.0.0.1 modem-script .*-v32 system-script cisco 1234
```

Modem-Specific Chat Script Examples

The following example shows line chat scripts being specified for lines connected to Telebit and US Robotics modems:

```
! Some lines have Telebit modems.
line 1 6
 script dialer telebit.*
! Some lines have US Robotics modems.
line 7 12
 script dialer usr.*
```

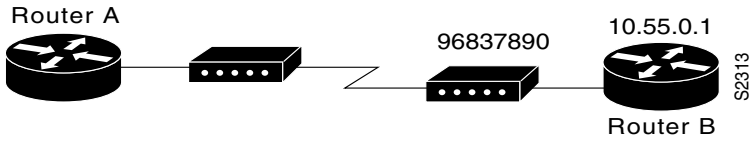
Dialer Mapping Example

The following example shows a modem chat script called dial and a system login chat script called login:

```
chat-script dial ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 60 CONNECT \c
chat-script login ABORT invalid TIMEOUT 60 name: myname word: mypassword ">" "slip
default"
interface async 10
 dialer in-band
 dialer map ip 10.55.0.1 modem-script dial system-script login 96837890
```

Figure 30 illustrates the configuration.

Figure 30 Chat Script Configuration and Function



- The configuration is on Router A.
- The modem chat script dial is used to dial out to the modem at Router B.
- The system login chat script login is used to log in to Router B.
- The phone number is the number of the modem attached to Router B.
- The IP address in the **dialer map** command is the address of Router B.

In the sample script shown, the **dialer in-band** command enables DDR on asynchronous interface 10, and the **dialer map** command dials 96837890 after finding the specified dialing and the system login scripts. When a packet is received for 10.55.0.1, the first thing to happen is that the modem script is implemented. Table 17 lists the functions that are implemented with each expect-send pair in the modem script called dial.

Table 17 Example Modem Script Execution

Expect and Send Pair	Implementation
ABORT ERROR	Ends the script execution if the text “ERROR” is found. (You can have as many active abort entries as you like.)
“ ” “AT Z”	Without expecting anything, sends an “AT Z” command to the modem. (Note the use of quotation marks to allow a space in the send string.)
OK “ATDT \T	Waits to see “OK.” Sends “ATDT 96837890.”
TIMEOUT 60	Waits up to 60 seconds for next expect string.
CONNECT \c	Expects “connect,” but does not send anything. (Note that \c is effectively nothing; “ ” would have indicated nothing followed by a carriage return.)

After the modem script is successfully executed, the system login script is executed. Table 18 lists the functions that are executed with each expect-send pair in the system script called login.

Table 18 Example System Script Execution

Expect and Send Pair	Implementation
ABORT invalid	Ends the script execution if the message “invalid username or password” is displayed.
TIMEOUT 60	Waits up to 60 seconds.
name: *username*	Waits for “name:” and sends username. (Using just “name:” will help avoid any capitalization issues.)
word: *password*	Waits for “word:” and sends the password.
“>” “slip default”	Waits for the > prompt and places the line into Serial Line Internet Protocol (SLIP) mode with its default address.

System Login Scripts and Modem Script Examples

The following example shows the use of chat scripts implemented with the **system-script** and **modem-script** options of the **dialer map** command.

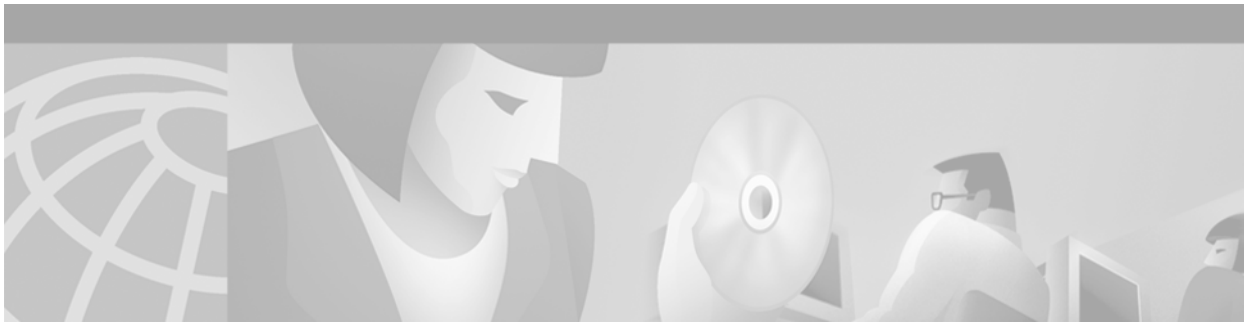
If there is traffic for IP address 10.2.3.4, the router will dial the 91800 number using the usrobotics-v32 script, matching the regular expression in the modem chat script. Then the router will run the unix-slip chat script as the system script to log in.

If there is traffic for 10.3.2.1, the router will dial 8899 using usrobotics-v32, matching both the modem script and modem chat script regular expressions. The router will then log in using the cisco-compressed script.

```
! Script for dialing a usr v.32 modem:
chat-script usrobotics-v32 ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 60 CONNECT \c
!
! Script for logging into a UNIX system and starting up SLIP:
chat-script unix-slip ABORT invalid TIMEOUT 60 name: billw word: wewpass ">" "slip
default"
!
```

```
! Script for logging into a Cisco access server and starting up TCP header compression:
chat-script cisco-compressed...
!
line 15
 script dialer usrobotics-*
!
interface async 15
 dialer map ip 10.2.3.4 system-script *-v32 system-script cisco-compressed 91800
 dialer map ip 10.3.2.1 modem-script *-v32 modem-script cisco-compressed 91800
```

ISDN Configuration



Configuring ISDN BRI

This chapter describes tasks that are required to use an ISDN BRI line. It provides an overview of the ISDN technologies currently available and describes features that you can configure in an ISDN BRI circuit-switched internetworking environment. This information is included in the following main sections:

- ISDN Overview
- How to Configure ISDN BRI
- Monitoring and Maintaining ISDN Interfaces
- Troubleshooting ISDN Interfaces
- Configuration Examples for ISDN BRI

This chapter describes configuration of the ISDN BRI. See the chapter "Configuring ISDN PRI" for information about configuring the ISDN PRI.

This chapter does not address routing issues, dialer configuration, and dial backup. For information about those topics, see the chapters in the "Dial-on-Demand Routing Configuration" part of this publication.

For hardware technical descriptions and for information about installing the router interfaces, refer to the appropriate hardware installation and maintenance publication for your particular product.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the BRI commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

ISDN Overview

Basic ISDN service is described in the section "ISDN Service" in the chapter "Overview of Dial Interfaces, Controllers, and Lines." To summarize, Cisco IOS software supports both the ISDN BRI and the ISDN PRI.

ISDN BRI provides two bearer (B) channels, each capable of transferring voice or data at 64 kbps, and one 16 kbps data (D) signaling channel, which 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, but many users use only half the available bandwidth.

Figure 9 in the chapter "Overview of Dial Interfaces, Controllers, and Lines" illustrates the channel assignment for each ISDN type.

Requesting BRI Line and Switch Configuration from a Telco Service Provider

Before configuring ISDN BRI on your Cisco router, you must order a correctly configured ISDN line from your telecommunications service provider. This process varies from provider to provider on a national and international basis. However, some general guidelines follow:

- Ask for two channels to be called by one number.
- Ask for delivery of calling line identification. Providers sometimes call this CLI or automatic number identification (ANI).
- If the router will be the only device attached to the BRI, ask for point-to-point service and a data-only line.
- If the router will be attached to an ISDN bus (to which other ISDN devices might be attached), ask for point-to-multipoint service (subaddressing is required) and a voice-and-data line.

When you order ISDN service for switches used in North America, request the BRI switch configuration attributes specified in Table 19.

Table 19 North American ISDN BRI Switch Type Configuration Information

Switch Type	Configuration
DMS-100 BRI Custom	2 B channels for voice and data. 2 directory numbers assigned by service provider. 2 service profile identifiers (SPIDs) required; assigned by service provider. Functional signaling. Dynamic terminal endpoint identifier (TEI) assignment. Maximum number of keys = 64. Release key = no, or key number = no. Ringing indicator = no. EKTS = no. PVC = 2. Request delivery of calling line ID on Centrex lines. Set speed for ISDN calls to 56 kbps outside local exchange. Directory number 1 can hunt to directory number 2.

Table 19 North American ISDN BRI Switch Type Configuration Information (continued)

Switch Type	Configuration
5ESS Custom BRI	**For Data Only** 2 B channels for data. Point to point. Terminal type = E. 1 directory number (DN) assigned by service provider. MTERM = 1. Request delivery of calling line ID on Centrex lines. Set speed for ISDN calls to 56 kbps outside local exchange. **For Voice and Data** (Use these values only if you have an ISDN telephone connected.) 2 B channels for voice or data. Multipoint. Terminal type = D. 2 directory numbers assigned by service provider. 2 SPIDs required; assigned by service provider. MTERM = 2. Number of call appearances = 1. Display = No. Ringing/idle call appearances = idle. Autohold = no. Onetouch = no. Request delivery of calling line ID on Centrex lines. Set speed for ISDN calls to 56 kbps outside local exchange. Directory number 1 can hunt to directory number 2.
5ESS National ISDN (NI) BRI	Terminal type = A. 2 B channels for voice and data. 2 directory numbers assigned by service provider. 2 SPIDs required; assigned by service provider. Set speed for ISDN calls to 56 kbps outside local exchange. Directory number 1 can hunt to directory number 2.
EZ-ISDN 1	**For Voice and Data** • ISDN Ordering Code for Cisco 766/776 Series = Capability S • ISDN Ordering Code for Cisco 1604 Series = Capability R 2 B channels featuring alternate voice and circuit-switched data. Non-EKTS voice features include the following: • Flexible Calling • Call Forwarding Variable • Additional Call Offering • Calling Number Identification (includes Redirecting Number Delivery)

Interface Configuration

The Cisco IOS software also provides custom features for configuring the ISDN BRI interface that provide such capability as call screening, called party number verification, ISDN default cause code override, and for European and Australian customers, Dialed Number Identification Service (DNIS)-plus-ISDN-subaddress binding to allow multiple binds between a dialer profile and an ISDN B channel.

Dynamic Multiple Encapsulations

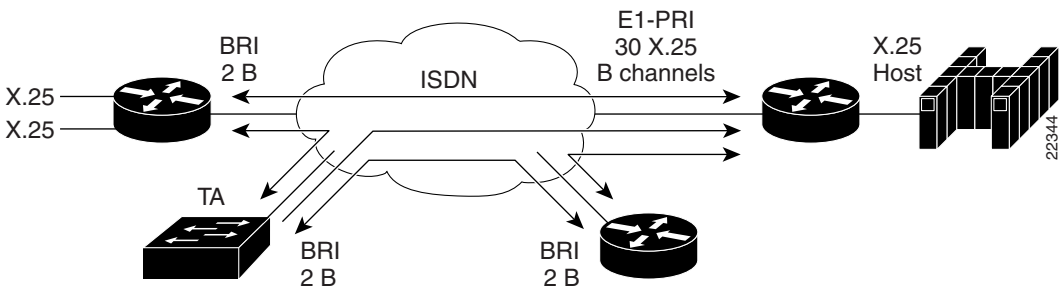
Before Cisco IOS Release 12.1, encapsulation techniques such as Frame Relay, High-Level Data Link Control (HDLC), Link Access Procedure, Balanced- Terminal Adapter (LAPB-TA), and X.25 could support only one ISDN B-channel connection over the entire link. HDLC and PPP could support multiple B channels, but the entire ISDN link needed to use the same encapsulation. The Dynamic Multiple Encapsulations feature introduced in Cisco IOS Release 12.1 allows various encapsulation types and per-user configurations on the same ISDN B channel at different times according to the type of incoming call.

With the Dynamic Multiple Encapsulations feature, once calling line identification (CLID) binding is completed, the topmost interface is always used for all configuration and data structures. The ISDN B channel becomes a forwarding device, and the configuration on the D channel is ignored, thereby allowing the different encapsulation types and per-user configurations. Dynamic multiple encapsulations provide support for packet assembler/disassembler (PAD) traffic and X.25 encapsulated and switched packets. For X.25 encapsulations, the configurations reside on the dialer profile.

Dynamic multiple encapsulation is especially important in Europe, where ISDN is relatively expensive and maximum use of all 30 B channels on the same ISDN link is desirable. Further, the feature removes the need to statically dedicate channels to a particular encapsulation and configuration type, and improves channel usage.

Figure 31 shows a typical configuration for an X.25 network in Europe. The Dynamic Multiple Encapsulations feature allows use of all 30 B channels, and supports calls that originate in diverse areas of the network and converge on the same ISDN PRI.

Figure 31 European X.25 Network



Interface Configuration Options

You can also optionally configure snapshot routing for ISDN interfaces. Snapshot routing is a method of learning remote routes dynamically and keeping the routes available for a specified period of time, even though routing updates are not exchanged during that period. See the chapter “Configuring Snapshot Routing” later in this guide for detailed information about snapshot routing.

To place calls on an ISDN interface, you must configure it with dial-on-demand routing (DDR). For configuration information about ISDN using DDR, see the "Dial-on-Demand Routing Configuration" part of this publication. For command information, refer to the *Cisco IOS Dial Technologies Command Reference*.

To configure bandwidth on demand, see the chapters "Configuring Legacy DDR Spokes" or "Configuring Legacy DDR Hubs" later in this publication.

ISDN Cause Codes

A cause code is an information element (IE) that indicates why an ISDN call failed or was otherwise disconnected. When the originating gateway receives a Release Complete message, it generates a tone corresponding to the cause code in the message.

Table 20 lists the default cause codes that the VoIP (Voice over IP) gateway sends to the switch when a call fails at the gateway, and the corresponding tones that it generates.

Table 20 Cause Codes Generated by the Cisco VoIP Gateway

Cause Code	Description	Explanation	Tone
1	Unallocated (unassigned) number	The ISDN number is not assigned to any destination equipment.	Reorder
3	No route to destination	The call was routed through an intermediate network that does not serve the destination address.	Reorder
16	Normal call clearing	Normal call clearing has occurred.	Dial
17	User busy	The called system acknowledged the connection request but was unable to accept the call because all B channels were in use.	Busy
19	No answer from user (user alerted)	The destination responded to the connection request but failed to complete the connection within the prescribed time. The problem is at the remote end of the connection.	Reorder
28	Invalid number format	The connection could not be established because the destination address was presented in an unrecognizable format or because the destination address was incomplete.	Reorder
34	No circuit/channel available	The connection could not be established because no appropriate channel was available to take the call.	Reorder

For a complete list of ISDN cause codes that are generated by the switch, refer to "Appendix B: ISDN Switch Types, Codes and Values" in the *Cisco IOS Debug Command Reference*.

Although the VoIP gateway generates the cause codes listed in Table 20 by default, there are commands introduced in previous Cisco IOS releases that can override these defaults, allowing the gateway to send different cause codes to the switch. The following commands override the default cause codes:

- **isdn disconnect-cause**—Sends the specified cause code to the switch when a call is disconnected.
- **isdn network-failure-cause**—Sends the specified cause code to the switch when a call fails because of internal network failures.
- **isdn voice-call-failure**—Sends the specified cause code to the switch when an inbound voice call fails with no specific cause code.

When you implement these commands, the configured cause codes are sent to the switch; otherwise, the default cause codes of the voice application are sent. For a complete description of these commands, refer to the *Cisco IOS Dial Technologies Command Reference*.

How to Configure ISDN BRI

To configure ISDN lines and interfaces, perform the tasks in the following sections:

- Configuring the ISDN BRI Switch (Required)
- Specifying Interface Characteristics for an ISDN BRI (As required)
- Configuring ISDN Semipermanent Connections (As required)
- Configuring ISDN BRI for Leased-Line Service (As required)

See the sections "Monitoring and Maintaining ISDN Interfaces" and "Troubleshooting ISDN Interfaces" later in this chapter for tips on maintaining your network. See the section "Configuration Examples for ISDN BRI" at the end of this chapter for configuration examples.

To configure ISDN BRI for voice, video, and fax applications, refer to the *Cisco IOS Voice, Video, and Fax Applications Configuration Guide*.

Configuring the ISDN BRI Switch

To configure the ISDN switch type, perform the following tasks:

- Configuring the Switch Type (Required)
- Checking and Setting the Buffers (As required)

Also see to the "Multiple ISDN Switch Types Feature" section for information about configuring multiple switch types.

Configuring the Switch Type

To configure the switch type, use the following command in global configuration mode:

Command	Purpose
`Router(config)# isdn switch-type switch-type`	Selects the service provider switch type; see Table 19 for switch types.

The section "Global ISDN and BRI Interface Switch Type Example" later in this chapter provides an example of configuring the ISDN BRI switch.

Table 21 lists the ISDN BRI service provider switch types.

Table 21 ISDN Service Provider BRI Switch Types

Switch Type Keywords	Description/Use	Central Office (CO) Switch Type?
Voice/PBX Systems		
basic-qsig	PINX (PBX) switch with QSIG signaling per Q.931	
Australia, Europe, and UK		
basic-1tr6	German 1TR6 ISDN switch	Yes
basic-net3	NET3 ISDN BRI for Norway NET3, Australia NET3, and New Zealand NET3 switches; covers ETSI-compliant Euro-ISDN E-DSS1 signaling system	Yes
vn3	French VN3 ISDN BRI switch	Yes
Japan		
ntt	Japanese NTT ISDN BRI switch	
North America		
basic-5ess	Lucent (AT&T) basic rate 5ESS switch	Yes
basic-dms100	Nortel basic rate DMS-100 switch	Yes
basic-ni	National ISDN switch	Yes
All Users		
none	No switch defined	

Note The command parser will still accept the following switch type keywords: **basic-nwnet3**, **vn2**, and **basic-net3**; however, when viewing the NVRAM configuration, the **basic-net3** or **vn3** switch type keywords are displayed respectively.

Checking and Setting the Buffers

When configuring a BRI, after the system comes up, make sure enough buffers are in the free list of the buffer pool that matches the maximum transmission unit (MTU) of your BRI interface. If not, you must reconfigure buffers in order for the BRI interfaces to function properly.

To check the MTU size and the buffers, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# show interfaces bri number`	Displays the MTU size.
`Router# show buffers`	Displays the free buffers.

To configure the buffers and the MTU size, use the following commands in global configuration mode as needed:

Command	Purpose

`Router(config)# buffers big permanent number`
`Router(config)# buffers big max-free number`
`Router(config)# buffers big min-free number`
`Router(config)# buffers big initial number`
	Configures the buffers.

Multiple ISDN Switch Types Feature

The Cisco IOS software provides an enhanced Multiple ISDN Switch Types feature that allows you to apply an ISDN switch type to a specific ISDN interface and configure more than one ISDN switch type per router. This feature allows both ISDN BRI and ISDN PRI to run simultaneously on platforms that support both interface types. See the section "Configuring Multiple ISDN Switch Types" in the chapter "Configuring ISDN PRI" for information about configuring this feature.

Specifying Interface Characteristics for an ISDN BRI

Perform the tasks in the following sections to set interface characteristics for an ISDN BRI, whether it is the only BRI in a router or is one of many. Each of the BRIs can be configured separately.

- Specifying the Interface and Its IP Address (Required)
- Configuring CLI Screening (As Required)
- Configuring Encapsulation on ISDN BRI (Required)
- Configuring Network Addressing (Required)
- Configuring TEI Negotiation Timing (Optional)
- Configuring CLI Screening (Optional)
- Configuring Called Party Number Verification (Optional)
- Configuring ISDN Calling Number Identification (Optional)
- Configuring the Line Speed for Calls Not ISDN End to End (Optional)
- Configuring a Fast Rollover Delay (Optional)
- Overriding ISDN Application Default Cause Codes (Optional)
- Configuring Inclusion of the Sending Complete Information Element (Optional)
- Configuring DNIS-plus-ISDN-Subaddress Binding (Optional)
- Screening Incoming V.110 Modem Calls (Optional)
- Disabling V.110 Padding (Optional)

Specifying the Interface and Its IP Address

To specify an ISDN BRI and enter interface configuration mode, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# interface bri number`
	Cisco 7200 series router only
	`Router(config)# interface bri slot/port`
		Specifies the interface and begins interface configuration mode.

Step 2	`Router(config-if)# ip address address mask`	Specifies an IP address for the interface.

Specifying ISDN SPIDs

Some service providers use SPIDs to define the services subscribed to by the ISDN device that is accessing the ISDN service provider. The service provider assigns the ISDN device one or more SPIDs when you first subscribe to the service. If you are using a service provider that requires SPIDs, your ISDN device cannot place or receive calls until it sends a valid, assigned SPID to the service provider when accessing the switch to initialize the connection.

Currently, only the DMS-100 and NI switch types require SPIDs. The AT&T 5ESS switch type may support a SPID, but we recommend that you set up that ISDN service without SPIDs. In addition, SPIDs have significance at the local access ISDN interface only. Remote routers never receive the SPID.

A SPID is usually a seven-digit telephone number with some optional numbers. However, service providers may use different numbering schemes. For the DMS-100 switch type, two SPIDs are assigned, one for each B channel.

To define the SPIDs and the local directory number (LDN) on the router, use the following commands in interface configuration mode as needed:

Command	Purpose
`Router(config-if)# isdn spid1 spid-number [ldn]`	Specifies a SPID and local directory number for the B1 channel.
`Router(config-if)# isdn spid2 spid-number [ldn]`	Specifies a SPID and local directory number for the B2 channel.

The LDN is optional but might be necessary if the router is to answer calls made to the second directory number.

Configuring Encapsulation on ISDN BRI

Each ISDN B channel is treated as a synchronous serial line, and the default serial encapsulation is HDLC. The Dynamic Multiple Encapsulations feature allows incoming calls over ISDN to be assigned an encapsulation type such as Frame Relay, PPP, and X.25 based on CLID or DNIS. PPP encapsulation is configured for most ISDN communication.

To configure encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation [ppp | lapb | frame-relay]`	Configures encapsulation type.

Verifying the Dynamic Multiple Encapsulations Feature

To verify dialer interfaces configured for binding and see statistics on each physical interface bound to the dialer interface, use the **show interfaces** EXEC command.

The following example shows that the output under the B channel keeps all hardware counts that are not displayed under any logical or virtual access interface. The line in the report that states "Interface is bound to Dialer0 (Encapsulation LAPB)" indicates that this B interface is bound to the dialer 0 interface and the encapsulation running over this connection is LAPB, not PPP, which is the encapsulation configured on the D interface and inherited by the B channel.

```
Router# show interfaces bri0:1

BRI0:1 is up, line protocol is up
  Hardware is BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  Interface is bound to Dialer0 (Encapsulation LAPB)
  LCP Open, multilink Open
  Last input 00:00:31, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 1 packets/sec
     110 packets input, 13994 bytes, 0 no buffer
     Received 91 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     135 packets output, 14175 bytes, 0 underruns
     0 output errors, 0 collisions, 12 interface resets
     0 output buffer failures, 0 output buffers swapped out
     8 carrier transitions
```

Any protocol configuration and states should be displayed from the dialer 0 interface.

Encapsulation Configuration Notes

The router might need to communicate with devices that require a different encapsulation protocol or the router might send traffic over a Frame Relay or X.25 network. The Dynamic Multiple Encapsulations feature provides bidirectional support of all serial encapsulations except Frame Relay.

For more information, see the sections "Sending Traffic over Frame Relay, X.25, or LAPB Networks" in the chapters "Configuring Legacy DDR Spokes" and "Configuring Legacy DDR Hubs" later in this publication.

To configure the router for automatic detection of encapsulation type on incoming calls, or to configure encapsulation for Cisco 700 and 800 series (formerly Combinet) router compatibility, see the section "Configuring Automatic Detection of Encapsulation Type" in the chapter "Configuring ISDN Special Signaling" later in this publication.

Configuring Network Addressing

The steps in this section support the primary goals of network addressing:

- Define which packets are *interesting* and will thus cause the router to make an outgoing call.
- Define the remote host where the calls are going.
- Specify whether broadcast messages will be sent.
- Specify the dialing string to use in the call.

Intermediate steps that use shared argument values tie the host identification and dial string to the interesting packets to be sent to that host.

To configure network addressing, use the following commands beginning in interface configuration mode:

	Command	Purpose

Step 1	`Router(config-if)# dialer map protocol next-hop-address name hostname speed [56 | 64] dial-string[:isdn-subaddress]`
	
	or
	
	`Router(config-if)# dialer map protocol next-hop-address name hostname spc [speed 56 | 64] [broadcast] dial-string[:isdn-subaddress]`
		(Most locations) Configures a serial interface or ISDN interface to call one or multiple sites or to receive calls from multiple sites.
		
		(Germany) Uses the command keyword that enables ISDN semipermanent connections.

Step 2	`Router(config-if)# dialer-group group-number`	Assigns the interface to a dialer group to control access to the interface.
Step 3	`Router(config-if)# exit`	Exits to global configuration mode.
Step 4	`Router(config)# dialer-list dialer-group protocol protocol-name {permit | deny | list access-list-number | access-group}`	Defines a dial-on-demand routing (DDR) dialer list for dialing by protocol or by a combination of a protocol and an access list.
Step 5	`Router(config)# access-list access-list-number {deny | permit} protocol source address source-mask destination destination-mask`	Defines an access list permitting or denying access to specified protocols, sources, or destinations. Permitted packets cause the router to place a call to the destination protocol address.

German networks allow semipermanent connections between customer routers with BRIs and the 1TR6 basic rate switches in the exchange. Semipermanent connections are less expensive than leased lines.

Note The access list reference in Step 5 of this task is an example of the **access-list** commands allowed by different protocols. Some protocols might require a different command form or might require multiple commands. Refer to the relevant protocol chapter in the network protocol configuration guide (the *Cisco IOS Novell IPX Configuration Guide*, for example) for more information about setting up access lists for a protocol.

For more information about defining outgoing call numbers, see the chapters “Configuring Legacy DDR Hubs” and “Configuring Legacy DDR Spokes” later in this publication.

Configuring TEI Negotiation Timing

You can configure ISDN TEI negotiation on individual ISDN interfaces. TEI negotiation is useful for switches that may deactivate Layers 1 or 2 when there are no active calls. Typically, this setting is used for ISDN service offerings in Europe and connections to DMS-100 switches that are designed to initiate TEI negotiation.

By default, TEI negotiation occurs when the router is powered up. The TEI negotiation value configured on an interface overrides the default or global TEI value. For example, if you configure **isdn tei first-call** globally and **isdn tei powerup** on BRI interface 0, then TEI negotiation **powerup** is the value applied to BRI interface 0. It is not necessary to configure TEI negotiation unless you wish to override the default value (**isdn tei powerup**).

To apply TEI negotiation to a specific BRI interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn tei [first-call | powerup]`	Determines when ISDN TEI negotiation occurs.

Configuring CLI Screening

CLI screening adds a level of security by allowing you to screen incoming calls. You can verify that the calling line ID is from an expected origin. CLI screening requires a local switch that is capable of delivering the CLI to the router.

To configure CLI screening, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn caller number`	Configures caller ID screening.

Note If caller ID screening is configured and the local switch does not deliver caller IDs, the router rejects all calls.

Note In earlier releases of the Cisco IOS software, ISDN accepted all synchronous calls and performed some minimal CLI screening before accepting or rejecting a call. Beginning with Cisco IOS Release 12.1 software, DDR provides a separate process that screens for the profile of the caller. The new screening process also checks that enough resources are available to accept the call and that the call conforms to predetermined rules. When the call is found acceptable, the screening process searches for a matching profile for the caller. The call is accepted only when there is a matching profile.

Configuring Called Party Number Verification

When multiple devices are attached to an ISDN BRI, you can ensure that only a single device answers an incoming call by verifying the number or subaddress in the incoming call against the configured number or subaddress or both of the device.

You can specify that the router verify a called-party number or subaddress number in the incoming setup message for ISDN BRI calls, if the number is delivered by the switch. You can do so by configuring the number that is allowed. To configure verification, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn answer1 [called-party-number][:subaddress]`	Specifies that the router verify a called-party number or subaddress number in the incoming setup message.

Verifying the called-party number ensures that only the desired router responds to an incoming call. If you want to allow an additional number for the router, you can configure it, too.

To configure a second number to be allowed, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn answer2 [called-party-number][:subaddress]`	Specifies that the router verify a second called-party number or subaddress number in the incoming setup message.

Configuring ISDN Calling Number Identification

A router with an ISDN BRI interface might need to supply the ISDN network with a billing number for outgoing calls. Some networks offer better pricing on calls in which the number is presented. When configured, this information is included in the outgoing call Setup message.

To configure the interface to identify the billing number, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn calling-number calling-number`	Specifies the calling party number.

This command can be used with all switch types except German 1TR6 ISDN BRI switches.

Configuring the Line Speed for Calls Not ISDN End to End

When calls are made at 56 kbps but delivered by the ISDN network at 64 kbps, the incoming data can be corrupted. However, on ISDN calls, if the receiving side is informed that the call is not an ISDN call from end to end, it can set the line speed for the incoming call.

To set the speed for incoming calls recognized as not ISDN end to end, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn not-end-to-end {56 | 64}`	Sets the speed to be used for incoming calls recognized as not ISDN end to end.

Configuring a Fast Rollover Delay

Sometimes a router attempts to dial a call on an ISDN B channel before a previous call is completely torn down. The fast rollover fails because the second call is made to a different number before the B channel is released from the unsuccessful call. This failure might occur in the following ISDN configurations:

- The two B channels of the BRI are not configured as a hunt group, but have separate numbers defined.
- The B channel is not released by the ISDN switch until after Release Complete signal is processed.

You need to configure this delay if a BRI on a remote peer has two phone numbers configured one for each B channel you are dialing into this BRI, you have a dialer map for each phone number, and the first call succeeds but a second call fails with no channel available.

To configure a fast rollover delay, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn fast-rollover-delay` *seconds*	Defines a fast rollover delay.

A delay of 5 seconds should cover most cases. Configure sufficient delay to make sure the ISDN RELEASE_COMPLETE message has been sent or received before making the fast rollover call. Use the **debug isdn q931** command to display this information. This pattern of failed second calls is a rare occurrence.

Overriding ISDN Application Default Cause Codes

The ISDN Cause Code Override function is useful for overriding the default cause code of ISDN applications. When this feature is implemented, the configured cause code is sent to the switch; otherwise, default cause codes of the application are sent.

To configure ISDN cause code overrides, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn disconnect-cause` {*cause-code-number* | **busy** | **not-available**}	Specifies the ISDN cause code to send to the switch.

ISDN Cause Code Override Configuration Example

The following example sends a BUSY cause code to the switch when an application fails to complete the call:

```
interface serial 0:23
 isdn disconnect-cause busy
```

Verifying ISDN Cause Code Override

To verify that the ISDN Cause Code Override feature is operating correctly, enter the **debug q931** command. The **debug q931** command displays a report of any configuration irregularities.

Configuring Inclusion of the Sending Complete Information Element

In some geographic locations, such as Hong Kong and Taiwan, ISDN switches require that the Sending Complete information element be included in the outgoing Setup message to indicate that the entire number is included. This information element is generally not required in other locations.

To configure the interface to include the Sending Complete information element in the outgoing call Setup message, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn sending-complete`	Includes the Sending Complete information element in the outgoing call Setup message.

Configuring DNIS-plus-ISDN-Subaddress Binding

To configure DNIS-plus-ISDN-subaddress binding, use the following command in global configuration mode

Command	Purpose
`Router(config)# dialer called DNIS:subaddress`	Binds a DNIS to an ISDN subaddress.

Note This command allows multiple binds between a dialer profile and an ISDN B channel. The configuration requires an ISDN subaddress, which is used in Europe and Australia.

See the section "DNIS-plus-ISDN-Subaddress Binding Example" later in this chapter for a configuration example.

Screening Incoming V.110 Modem Calls

You can screen incomingV.110 modem calls and reject calls that do not have the communications settings configured as the network expects them to be.

To selectively accept incoming V.110 modem calls based on data bit, parity, and stop bit modem communications, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn v110 only [databits {5 | 7 | 8}] [parity {even | mark | none | odd | space}] [stopbits {1 | 1.5 | 2}]`	Selectively accepts incoming V.110 calls based on data bit, parity, and stop bit modem communication settings.

Disabling V.110 Padding

In networks with devices such as terminal adapters (TAs) and global system for mobile communication (GSM) handsets that do not fully conform to the V.110 modem standard, you will need to disable V.110 padding. To disable the padded V.110 modem speed report required by the V.110 modem standard, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# no isdn v110 padding`	Disables the padded modem speed report required by the V.110 modem standard.

Configuring ISDN Semipermanent Connections

German networks allow semipermanent connections between customer routers with BRI interfaces and the 1TR6 basic rate switches in the exchange. Australian networks allow semipermanent connections between ISDN PRI interfaces and the TS-014 primary rate switches in the exchange. Semipermanent connections are offered at better pricing than leased lines.

Configuring BRI interfaces for semipermanent connection requires only that you use a keyword that indicates semipermanent connections when you are setting up network addressing as described in the previous section of this chapter.

To configure a BRI for semipermanent connections, follow this procedure:

Step 1 Set up the ISDN lines and ports as described in the sections "Configuring the ISDN BRI Switch"and "Specifying Interface Characteristics for an ISDN BRI" or for ISDN PRI, see the section "How to Configure ISDN PRI" in the chapter "Configuring ISDN PRI" later in this manual.

Step 2 Configure DDR on a selected interface, as described in the "Dial-on-Demand Routing Configuration" part of this publication.

To begin DDR network addressing, use the following command in interface configuration mode

:

Command	Purpose
`Router(config-if)# dialer map` *protocol next-hop-address* `name` hostname `spc` [`speed 56` | `64`] [`broadcast`] *dial-string*[`:`*isdn-subaddress*]	Defines the remote recipient's protocol address, host name, and dialing string; indicates semipermanent connections; optionally, provides the ISDN subaddress; and sets the dialer speed to 56 or 64 kbps, as needed.

Configuring ISDN BRI for Leased-Line Service

To configure ISDN BRI for leased line service, perform the tasks in one of the following sections as needed and available:

- Configuring Leased-Line Service at Normal Speeds (Available in Japan and Germany)
- Configuring Leased-Line Service at 128 Kbps (Available only in Japan)

Note Once an ISDN BRI interface is configured for access over leased lines, it is no longer a dialer interface, and signaling over the D channel no longer applies. Although the interface is called **interface bri** *n*, it is configured as a synchronous serial interface having the default High-Level Data Link (HDLC) encapsulation. However, the Cisco IOS commands that set the physical characteristics of a serial interface (such as the pulse time) do not apply to this interface.

Configuring Leased-Line Service at Normal Speeds

This service is offered in Japan and Germany and no call setup or teardown is involved. Data is placed on the ISDN interface similar to the way data is placed on a leased line connected to a serial port.

To configure the BRI to use the ISDN connection as a leased-line service, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type switch-type`	Configures the BRI switch type, as specified by the local service provider.
Step 2	`Router(config)# isdn leased-line bri number 128`	Specifies the BRI interface number.

To disable leased-line service if you no longer want to support it on a specified ISDN BRI, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no isdn leased-line bri number`	Removes leased line configuration from a specified ISDN BRI interface.

Configuring Leased-Line Service at 128 Kbps

The Cisco IOS software supports leased-line service at 128 kbps via ISDN BR. This service combines two B channels into a single pipe. This feature requires one or more ISDN BRI hardware interfaces that support channel aggregation and service provider support for ISDN channel aggregation at 128 kbps. When this software first became available, service providers offered support for ISDN channel aggregation at 128 kbps only in Japan.

Note This feature is not supported on the Cisco 2500 series router because its BRI hardware does not support channel aggregation.

To enable leased-line service at 128 kbps on a specified ISDN BRI, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type switch-type`	Selects the service provider switch type.
Step 2	`Router(config)# isdn leased-line bri number 128`	Configures a specified BRI for access over leased lines.

To complete the configuration of the interface, see the chapter "Configure a Synchronous Serial Ports" in this publication.

To remove the leased-line service configuration from a specified ISDN BRI, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no isdn leased-line bri` *number*	Removes leased-line configuration from a specified ISDN BRI interface.

Monitoring and Maintaining ISDN Interfaces

To monitor and maintain ISDN interfaces, use the following commands in EXEC mode as needed:

Command	Purpose
`Router> show interfaces bri` *number* **Cisco 7200 series routers only** `Router> show interfaces bri` *slot*`/`*port*	Displays information about the physical attributes of the ISDN BRI B and D channels.
`Router> show controllers bri` *number* **Cisco 7200 series routers only** `Router> show controllers bri` *slot*`/`*port*	Displays protocol information about the ISDN B and D channels.
`Router> show isdn {active | history | memory | status | timers}`	Displays information about calls, history, memory, status, and Layer 2 and Layer 3 timers.
`Router> show dialer interface bri` *number*	Obtains general diagnostic information about the specified interface.

Troubleshooting ISDN Interfaces

To test the ISDN configuration of the router, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# show controllers bri` *number*	Checks Layer 1 (physical layer) of the BRI.
`Router# debug q921`	Checks Layer 2 (data link layer).

`Router# debug isdn events`
`Router# debug q931`
`Router# debug dialer`
`Router# show dialer`
	Checks Layer 3 (network layer).

Refer to the *Cisco IOS Debug Command Reference* for more information about the **debug** commands.

Configuration Examples for ISDN BRI

This section provides the following ISDN BRI configuration examples:

Global ISDN and BRI Interface Switch Type Example

The following example shows a global National ISDN switch type (keyword **basic-ni**) and an interface-level NET3 ISDN switch type (keyword **basic-net3**). The **basic-net3** keyword is applied to BRI interface 0 and overrides the global switch setting.

```
isdn switch-type basic-ni
!
interface BRI0
 isdn switch-type basic-net3
```

BRI Connected to a PBX Example

The following example provides a simple partial configuration of a BRI interface that is connected to a PBX. This interface is connected to a switch that uses SPID numbers.

```
interface BRI0
 description connected to pbx line 61885
 ip address 10.1.1.3 255.255.255.0
 encapsulation ppp
 isdn spid1 123
 dialer map ip 10.1.1.1 name mutter 61886
 dialer map ip 10.1.1.2 name rudder 61884
 dialer map ip 10.1.1.4 name flutter 61888
 dialer-group 1
 no fair-queue
 ppp authentication chap
```

Multilink PPP on a BRI Interface Example

The following example enables Multilink PPP on BRI 0:

```
interface BRI0
 description Enables PPP Multilink on BRI 0
 ip address 10.1.1.1 255.255.255.0
```

```
 encapsulation ppp
 dialer map ip 10.1.1.2 name coaster 14195291357
 dialer map ip 10.1.1.3 name roaster speed 56 14098759854
 ppp authentication chap
 ppp multilink
 dialer-group 1
```

Dialer Rotary Groups Example

The following example configures BRI interfaces to connect into a rotary group (using the **dialer-group** command) and then configures a dialer interface for that dialer group. This configuration permits IP packets to trigger calls.

```
interface BRI 0
 description connected into a rotary group
 encapsulation ppp
 dialer rotary-group 1

interface BRI 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1

interface BRI 2
 encapsulation ppp
 dialer rotary-group 1

interface BRI 3
 no ip address
 encapsulation ppp
 dialer rotary-group 1

interface BRI 4
 encapsulation ppp
 dialer rotary-group 1

interface Dialer 0
 description Dialer group controlling the BRIs
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name angus 14802616900
 dialer-group 1
 ppp authentication chap

dialer-list 1 protocol ip permit
```

Compression Examples

The following example enables predictor compression on BRI 0:

```
interface BRI0
 description Enables predictor compression on BRI 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name bon 14195291357
 compress predictor
 ppp authentication chap
 dialer-group 1
```

The following example enables stacker compression on BRI 0:

```
interface BRI0
```

```
 description Enables stac compression on BRI 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name malcom 14195291357
 compress stac
 ppp authentication chap
 dialer-group 1
```

Multilink PPP and Compression Example

The following example enables Multilink PPP and stacker compression on BRI 0:

```
interface BRI0
 description Enables PPP Multilink and stac compression on BRI 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name rudd 14195291357
 ppp authentication chap
 compress stac
 ppp multilink
 dialer-group 1
```

Voice over ISDN Examples

The following example allows incoming voice calls to be answered on BRI 0:

```
interface bri0
 description Allows incoming voice calls to be answered on BRI 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 isdn incoming-voice data
 dialer map ip 10.1.1.2 name starstruck 14038182344
 ppp authentication chap
 dialer-group 1
```

The following example allows outgoing voice calls on BRI 1:

```
interface bri1
 description Places an outgoing call as a voice call on BRI 1
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name angus class calltype 19091238877
 ppp authentication chap
 dialer-group 1

map-class dialer calltype
 dialer voice-call
```

For more configuration examples of voice calls over ISDN, refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide*.

DNIS-plus-ISDN-Subaddress Binding Example

The following example configures a dialer profile for a receiver with DNIS 12345 and ISDN subaddress 6789:

```
dialer called 12345:6789
```

For additional configuration examples, see the sections "Dynamic Multiple Encapsulations" and "Verifying the Dynamic Multiple Encapsulations Feature" in the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles" in this publication.

Screening Incoming V.110 Modem Calls Example

The following example filters out all V.110 modem calls except those with communication settings of 8 data bits, no parity bit, and 1 stop bit:

```
interface serial 0:23
  isdn v110 only databits 8 parity none stopbits 1
```

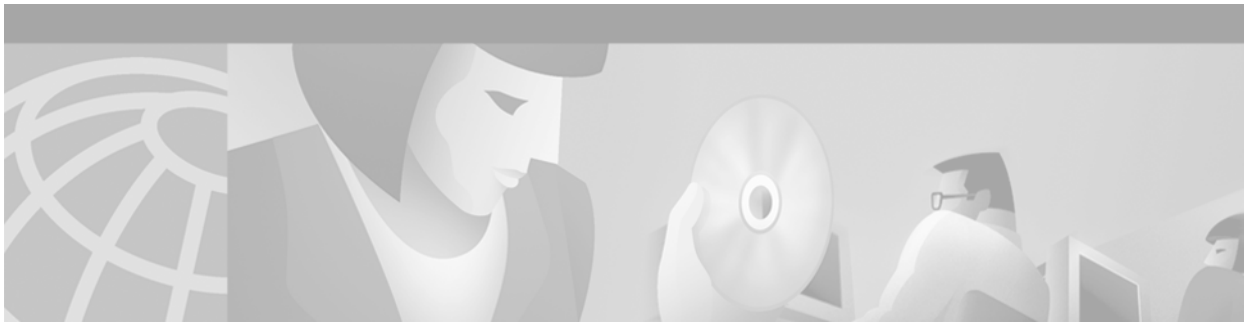
ISDN BRI Leased-Line Configuration Example

The following example configures the BRI 0 interface for leased-line access at 128 kbps. Because of the leased-line–not dialed–environment, configuration of ISDN called and calling numbers are not needed and not used. The BRI 0 interface is henceforth treated as a synchronous serial interface, with the default HDLC encapsulation.

```
isdn leased-line bri 0 128
```

The following example configures the BRI 0 interface for PPP encapsulation:

```
interface bri 0
 ip address 10.1.1.2 255.255.255.0
 encapsulation ppp
 bandwidth 128
```



Configuring Virtual Asynchronous Traffic over ISDN

Cisco IOS software offers two solutions to send virtual asynchronous traffic over ISDN:

- Using International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Recommendation V.120, which allows for reliable transport of synchronous, asynchronous, or bit transparent data over ISDN bearer channels.
- Using ITU-T Recommendation X.75, which allows a system with an ISDN terminal adapter supporting asynchronous traffic over Link Access Procedure, Balanced (LAPB) to call into a router and establish an asynchronous PPP session. This method of asynchronous traffic transmission is also called ISDN Link Access Procedure, Balanced-Terminal Adapter (LAPB-TA).

A virtual asynchronous interface (also known as vty-async) is created on demand to support calls that enter the router through a nonphysical interface. For example, asynchronous character stream calls terminate or land on nonphysical interfaces. These types of calls include inbound Telnet, local-area transport (LAT), PPP over character-oriented protocols (such as V.120 or X.25), and LAPB-TA and packet assembler/disassembler (PAD) calls.

Virtual asynchronous interfaces are not user configurable; rather, they are dynamically created and torn down on demand. A virtual asynchronous line is used to access a virtual asynchronous interface. Refer to the section "Virtual Asynchronous Interfaces" in the chapter "Overview of Dial Interfaces, Controllers, and Lines" in this publication for more overview information about virtual asynchronous interfaces. Refer to the section "Enabling Asynchronous Functions on Virtual Terminal Lines" in the chapter "Configuring Protocol Translation and Virtual Asynchronous Devices" in the Cisco *IOS Terminal Services Configuration Guide*, for additional virtual asynchronous interface configuration information.

This chapter describes how to configure virtual asynchronous traffic over ISDN lines. It includes the following main sections:

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Recommendation V.120 Overview

The V-series recommendations are ITU-T standards dealing with data communications over telephone networks. V.120 allows for reliable transport of synchronous, asynchronous, or bit transparent data over ISDN bearer channels. Cisco provides three V.120 support features for terminal adapters that do not send the low-layer compatibility fields or bearer capability V.120 information:

- Answer all incoming calls as V.120—Static configuration used when all remote users have asynchronous terminals and need to connect with a vty on the router.
- Automatically detect V.120 encapsulation—Encapsulation dynamically detected and set.
- Enable V.120 Support for Asynchronous Access over ISDN.

For terminal adapters that send the low-layer compatibility or bearer capability V.120 information, mixed V.120 and ISDN calls are supported. No special configuration is required.

How to Configure V.120 Access

To configure V.120 access, perform the tasks in the following sections:

- Configuring Answering of All Incoming Calls as V.120 (Required)
- Configuring Automatic Detection of Encapsulation Type (Required)
- Enabling V.120 Support for Asynchronous Access over ISDN (Required)

See the section "Configuration Example for V.120" at the end of this chapter for an example of how to configure V.120 access.

Configuring Answering of All Incoming Calls as V.120

This V.120 support feature allows users to connect using an asynchronous terminal over ISDN terminal adapters with V.120 support to a vty on the router, much like a direct asynchronous connection. Beginning with Cisco IOS Release 11.1, this feature supports incoming calls only.

When all the remote users have asynchronous terminals and call in to a router through an ISDN terminal adapter that uses V.120 encapsulation but does not send the low-layer compatibility or bearer capability V.120 information, you can configure the interface to answer all calls as V.120. Such calls are connected with an available vty on the router.

To configure an ISDN BRI or PRI interface to answer all incoming calls as V.120, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco 4000 series routers only**
	`Router(config)# interface bri number`
	or
	Cisco 7200 series routers only
	`Router(config)# interface bri slot/port`
		Configures the ISDN BRI interface and begins interface configuration mode.

Step 2	`Router(config)# interface serial e1 controller-number:15`
	or
	`Router(config)# interface serial t1 controller-number:23`
		Configures the ISDN PRI D channel and begins interface configuration mode.

Step 3	`Router(config-if)# isdn all-incoming-calls-v120`	Configures the interface to answer all calls as V.120.

Configuring Automatic Detection of Encapsulation Type

If an ISDN call does not identify the call type in the lower-layer compatibility fields and is using an encapsulation that is different from the one configured on the interface, the interface can change its encapsulation type dynamically.

This feature enables interoperation with ISDN terminal adapters that use V.120 encapsulation but do not signal V.120 in the call setup message. An ISDN interface that by default answers a call as synchronous serial with PPP encapsulation can change its encapsulation and answer such calls.

Automatic detection is attempted for the first 10 seconds after the link is established or the first 5 packets exchanged over the link, whichever is first.

To enable automatic detection of V.120 encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# autodetect encapsulation v120`	Enables automatic detection of encapsulation type on the specified interface.

You can specify one or more encapsulations to detect. Cisco IOS software currently supports automatic detection of PPP and V.120 encapsulations.

Enabling V.120 Support for Asynchronous Access over ISDN

You can optionally configure a router to support asynchronous access over ISDN by globally enabling PPP on vty lines. Asynchronous access is then supported over ISDN from the ISDN terminal to the vty session on the router.

To enable asynchronous protocol features on vty lines, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vty-async`	Configures all vty lines to support asynchronous protocol features.

This task enables PPP on vty lines on a global basis on the router. If you prefer instead to configure PPP on a per-vty basis, use the **translate** command, which is described in the *Cisco IOS Dial Technologies Command Reference*.

Configuration Example for V.120

The following example configures BRI 0 to call and receive calls from two sites, to use PPP encapsulation on outgoing calls, and to use Challenge Handshake Authentication Protocol (CHAP) authentication on incoming calls. This example also enables BRI 0 to configure itself dynamically to answer calls that use V.120 but that do not signal V.120 in the call setup message.

```
interface bri 0
 encapsulation ppp
 autodetect encapsulation v120
 no keepalive
 dialer map ip 172.18.36.10 name EB1 234
 dialer map ip 172.18 36.9 name EB2 456
 dialer-group 1
ppp authentication chap
```

ISDN LAPB-TA Overview

To carry asynchronous traffic over ISDN, your system must be able to convert that traffic and forward it over synchronous connections. This process can be implemented by the V.120 protocol, which carries asynchronous traffic over ISDN. However, several countries in Europe (Germany, Switzerland, and some Eastern European countries) use LAPB as the protocol to forward their asynchronous traffic over synchronous connections. Your system, therefore, must be able to recognize and accept calls from these asynchronous/synchronous conversion devices. LAPB-TA performs that function. (LAPB is sometimes referred to as "X.75," because LAPB is the link layer specified in the ITU-T X.75 recommendation for carrying asynchronous traffic over ISDN.)

LAPB-TA allows devices that use LAPB instead of the V.120 protocol to communicate with routers on the Cisco 3600 and 5300 series.

LAPB supports both local CHAP authentication and external RADIUS authorization on the authentication, authorization, and accounting (AAA) server.

Before configuring ISDN LAPB-TA in your network, observe these restrictions:

- LAPB-TA does not currently support the ability to set a maximum frame size per user.
- Outbound LAPB-TA calls are not supported.

- PPP over LAPB-TA (and V.120) connections impose a greater overhead on the router than synchronous PPP over ISDN. The number of simultaneous sessions can be limited by dedicating a pool of virtual terminals to these protocols and limiting the number of virtual terminals in the pool.
- Multilink PPP compression is not supported.

How to Configure ISDN LAPB-TA

ISDN LAPB-TA is supported on the Cisco 3600 and Cisco 5300 series routers that meet the following additional requirements:

- A virtual terminal must be configured for incoming LAPB-TA. If no appropriately configured virtual terminals are available, the incoming call will be cleared.
- ISDN, LAPB, and PPP must be running to configure LAPB-TA.
- The Cisco IOS software must include the **vty-async** global configuration command, which must be configured before you can run asynchronous PPP traffic over a LAPB-TA connection.

If an interface is already configured for V.120, only the following two additional configuration commands are required on the interface because V.120 and LAPB-TA sessions are configured in a similar way:

- Use the **autodetect encapsulation** command to enable autodetection of LAPB-TA connections.
- Use the **transport input** command to list LAPB-TA as an acceptable transport on a specific router.

Perform the following required task to configure LAPB-TA: To configure ISDN LAPB-TA, use the following commands beginning in global configuration command mode: (required).

Procedures for verifying the configuration are found in the section "Verifying ISDN LAPB-TA" later in this chapter. The section "Configuration Example for ISDN LAPB-TA" at the end of this chapter provides configuration examples.

To configure ISDN LAPB-TA, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# vty-async`	Creates a virtual asynchronous interface.
Step 2	`Router(config)# vty-async virtual-template 1`	Applies virtual template to the virtual asynchronous interface.
Step 3	`Router(config)# interface virtual-template 1`	Creates a virtual interface template and enters interface configuration mode.
Step 4	`Router(config-if)# ip unnumbered Ethernet0`	Assigns an IP address to the virtual interface template.
Step 5	`Router(config-if)# encapsulation ppp`	Enables encapsulation on the virtual interface template.
Step 6	`Router(config-if)# no peer default ip address`	Disables an IP address from a pool to the device connecting to the virtual access interface
Step 7	`Router(config-if)# ppp authentication chap`	Enables the CHAP protocol for PPP authentication.
Step 8	`Router(config-if)# exit`	Exits to global configuration mode.

	Command	Purpose
Step 9	`Router(config)# username user1 password home`	Specifies CHAP password to be used to authenticate calls from caller "user1."
Step 10	`Router(config)# interface Serial0:236`	Enters interface configuration mode for a D-channel serial interface.[1]
Step 11	`Router(config-if)# encapsulation ppp`	Configures PPP encapsulation as the default.
Step 12	`Router(config-if)# dialer-group 1`	Specifies the dialer group belonging to the interface.
Step 13	`Router(config-if)# ppp authentication chap`	Enables the CHAP protocol for PPP authentication.
Step 14	`Router(config-if)# autodetect encapsulation lapb-ta`	Enables autodetect encapsulation for LAPB-TA protocols.
Step 15	`Router(config)# line vty 0 32`	Configures a range of 32 vty lines starting with vty0.
Step 16	`Router(config-line)# transport input telnet lapb-ta`	Defines which protocol to use to connect to a specific line of the access server.

1. The D channel is the signaling channel.

Verifying ISDN LAPB-TA

Enter the **show running configuration** command to verify that LAPB-TA is configured. The following output shows LAPB-TA enabled for serial interface 0:23:

```
Router# show running configuration

Building configuration...

Current configuration:
!
version 12.0
service timestamps debug datetime msec localtime
service timestamps log datetime msec localtime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname Router
...(output omitted)

interface Serial0:23
 description ENG PBX BRI num.:81063
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 dialer pool-member 1
 autodetect encapsulation ppp lapb-ta
 isdn switch-type primary-5ess
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp authentication chap
...(output omitted)
!
end
```

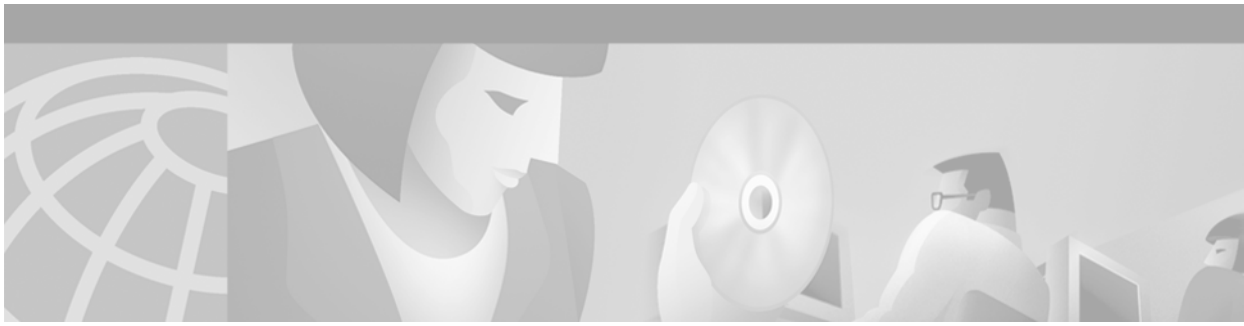
Configuration Example for ISDN LAPB-TA

The following example configures a virtual template LAPB-TA connection capable of running PPP. It assumes that you have already configured usernames and passwords for PPP authentication.

```
vty-async
vty-async virtual-template 1
interface virtual-template 1
 ip unnumbered Ethernet0
 encapsulation ppp
 no peer default ip address
 ppp authentication chap
 exit
 interface Serial0:23
 autodetect encapsulation lapb-ta
```

The following example treats the LAPB-TA and V.120 calls identically by immediately starting a PPP session without asking for username and password and relying on PPP authentication to identify the caller:

```
vty-async
vty-async virtual-template 1
interface Loopback0
 ip address 10.2.2.1 255.255.255.0
 exit
 interface BRI3/0
 encapsulation ppp
 autodetect encapsulation ppp lapb-ta v120
 exit
interface Virtual-Template1
 ip unnumbered Loopback0
 ppp authentication chap
 exit
ip local pool default 10.2.2.64 10.2.2.127
line vty 0 2
 password <removed>
 login
 transport input telnet
 exit
line vty 3 4
 no login
 transport input lapb-ta v120
 autocommand ppp neg
 exit
 end
```



Configuring Modem Use over ISDN BRI

This chapter describes how to configure the Modem over ISDN BRI feature. It includes the following main sections:

- Modem over ISDN BRI Overview
- How to Configure Modem over ISDN BRI
- Verifying ISDN BRI Interface Configuration
- Configuration Examples for Modem over ISDN BRI

Before beginning the tasks in this chapter, check your system for the following hardware and software:

- At least one of the following digital modem network modules. The number in the model name indicates the number of digital modems that can be connected to the module.
  - NM-6DM
  - NM-12DM
  - NM-18DM
  - NM-24DM
  - NM-30DM

  These digital modem network modules do not have their own network connections, but instead handle analog calls passing through other router interfaces. BRI modules can provide their ISDN connectivity. Other modules, such as Ethernet, can provide connectivity to the LAN. The digital modem module acts as a pool of available modems that can be used for both incoming and outgoing calls. Digital modem network modules *do not* support BRI voice interface cards or wide-area network (WAN) interface cards.
- At least one of the following Cisco BRI network modules:
  - NM-4B-S/T: 4-port ISDN BRI network module, minimum version 800-01236-03
  - NM-4B-U: 4-port ISDN BRI with integrated network termination 1 (NT-1) network module, minimum version 800-01238-06
  - NM-8B-S/T: 8-port ISDN BRI network module, minimum version 800-01237-03
  - NM-8B-U: 8-port ISDN BRI with integrated NT-1 network module, minimum version 800-01239-06

  The version level is available from the **show diag** command, which displays the version number as the part number.

If your BRI network module is a version lower than those cited or you need more details, refer to the Cisco.com Field Notice titled *Using Digital Modems with the Cisco 3600 Basic Rate Interface (BRI) Network Module Upgrade* in the Access Products index. If your existing Cisco BRI network module is one of those listed and does not support the Modem over ISDN BRI feature, Cisco will upgrade the module at no charge.

- To support the Modem over ISDN BRI feature, V.90 modem portware—for instructions on downloading this software or obtaining it otherwise, refer to the *Cisco 3600 Series Modem Portware Upgrade Configuration Note* on Cisco.com.

Before you can configure a Cisco 3640 router to provide Modem over ISDN BRI connectivity, you must also perform the following tasks:

- Obtain BRI service from your telecommunications provider. The BRI line must be provisioned at the switch to support voice calls.
- Install a 4-port or 8-port BRI network module into your Cisco router. Depending on the type of network module and your BRI service, you might also need to install an external NT-1 for S/T interfaces.
- Install a supported digital modem network module into the Cisco 3640 router.
- After the system comes up, make sure enough buffers are in the free list of the buffer pool that matches the maximum transmission unit (MTU) of your BRI interface. If not, you must reconfigure buffers so the BRI interfaces function properly. To check the MTU of your interfaces, use the **show interfaces bri** command. The **show buffers** command displays the free buffer space. Use the **buffers** global configuration command to make adjustments to initial buffer pool settings and to the limits at which temporary buffers are created and destroyed.

For more information about the physical characteristics of the BRI network modules and their digital modem support, or instructions on how to install the network or modem modules, either refer to the Cisco 3600 series *Network Module Hardware Installation Guide* that came with your BRI network module or view the up-to-date information on CCO.

To identify the hardware platform or software image information associated with a feature, 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.

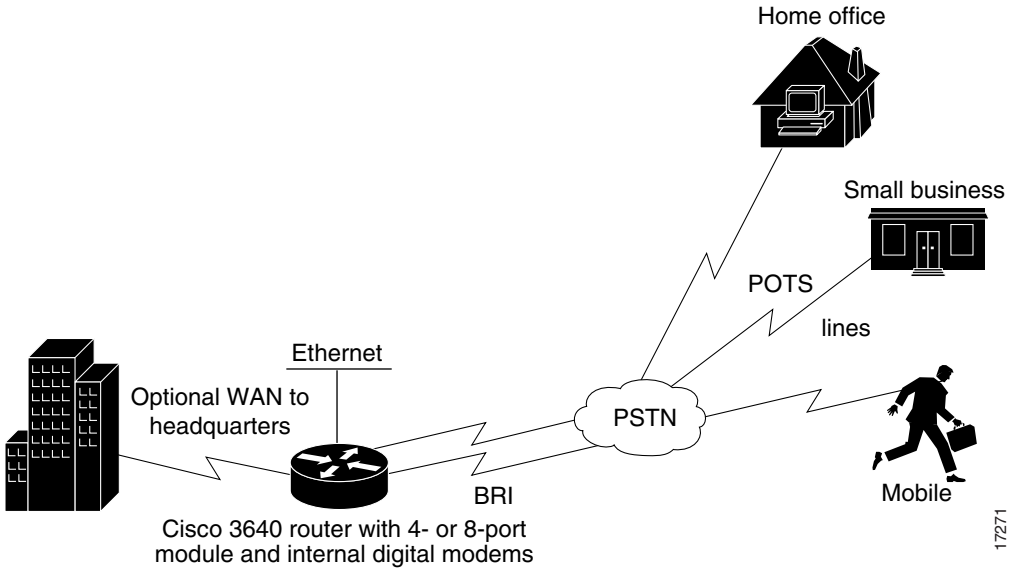
For a complete description of the Modem over ISDN BRI commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Modem over ISDN BRI Overview

The Modem over ISDN BRI feature for the Cisco 3640 modular access router lowers the cost of remote access by offering high-speed modem and ISDN connectivity for mobile customers, offices, and other remote-access users. Branch offices and enterprises can support analog modem users who call over the Public Switched Telephone Network (PSTN) into BRI interfaces in Cisco 3640 routers.

The digital modem in the router accepts the modem calls at connection speeds as fast as 56 kbps, adhering to the V.90 standard. As shown in Figure 32, the Cisco 3640 router in this way provides rapid access to E-mail and other network services.

Figure 32 Modem over ISDN BRI Feature



The following are benefits of using the Modem over ISDN BRI feature:

- Supports cost-effective and readily available BRI service.
- Provides remote modem users with rapid Internet and LAN/WAN access.
- Allows flexible remote access application support.

How to Configure Modem over ISDN BRI

The Modem over ISDN BRI feature is part of interface configuration for BRI. You configure the BRI interface after you have configured the ISDN global characteristics, which are switch type and TEI negotiation timing. These characteristics can also be defined for each BRI interface, as shown in the following task table.

To set up the BRI interface characteristics, set the global parameters and then configure each interface separately by using the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type switch-type`	Configures the global ISDN switch type to match the service provider switch type. For a list of keywords, see Table 22.
Step 2	`Router(config)# isdn tei [first-call | powerup]`	Configures when the ISDN TEI negotiation occurs. If this command is not used, negotiation occurs when the router is powered up. The **first-call** option is primarily used in European ISDN switch types, such as NET3 networks. The **powerup** option should be used in most other locations.

	Command	Purpose
Step 3	`Router(config)# interface bri slot/port`	Begins interface configuration mode to configure parameters for the specified interface. *slot* is the location of the BRI module. Valid values are from 0 to 3. *port* is an interface number. Valid values are from 0 to 7 if the module is an 8-port BRI network module, or from 0 to 4 if the module is a 4-port BRI network module.
Step 4	`Router(config-if)# ip address ip-address mask`	Specifies an IP address and subnet for the interface. You can also specify that there is no IP address. For information about IP addressing, see the Release 12.2 *Cisco IOS IP Configuration Guide* publication.
Step 5	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the BRI interface. PPP encapsulation is configured for most ISDN communication. If the router needs to communicate with devices that require a different encapsulation protocol, needs to detect encapsulation on incoming calls automatically, or needs to send traffic over a Frame Relay or X.25 network, see the chapter "Configuring X.25 on ISDN" later in this part, and the chapters in the Dial-on-Demand Routing Configuration part of this publication for information.

Step 6	`Router(config-if)# dialer map protocol next-hop-address name hostname speed 56|64 dial-string[:isdn-subaddress]`
	or
	`Router(config-if)# dialer map protocol next-hop-address name hostname spc [speed 56 | 64] [broadcast] dial-string[:isdn-subaddress]`
		(Most locations) Defines the remote protocol address of the recipient, host name, and dialing string; optionally, provide the ISDN subaddress; set the dialer speed to 56 or 64 kbps, as needed.
		(Germany) Use the **spc** keyword to enable ISDN semipermanent connections.

Step 7	`Router(config-if)# dialer-group group-number`	Assigns the interface to a dialer group to control access to the interface.
Step 8	`Router(config-if)# dialer-list dialer-group list access-list-number`	Associates the dialer group number with an access list number.
Step 9	`Router(config-if)# access-list access-list-number {deny | permit} protocol source address source-mask destination destination-mask`	Defines an access list permitting or denying access to specified protocols, sources, or destinations. Permitted packets cause the router to place a call to the destination protocol address.
Step 10	`Router(config-if)# no ip-directed broadcast`	(Optional) Disables the translation of directed broadcast to physical broadcasts.
Step 11	`Router(config-if)# isdn switch-type switch-type`	(Optional) Configures the interface ISDN switch type to match the service provider switch type. The interface ISDN switch type overrides the global ISDN switch type on the interface. For a list of keywords, refer to Table 22.

	Command	Purpose
Step 12	`Router(config-if)# isdn tei [first-call | powerup]`	(Optional) Determines when ISDN TEI negotiation occurs for an individual interface. This overrides the global configuration command.
Step 13	`Router(config-if)# isdn spid1 spid-number [ldn]`	Specifies a service profile identifier (SPID) and local directory number for the B1 channel. Currently, only the DMS-100 and NI-1 switch types require SPIDs. Although the Lucent 5ESS switch type might support a SPID, we recommend that you set up that ISDN service without SPIDs.
Step 14	`Router(config-if)# isdn spid2 spid-number [ldn]`	Specifies a SPID and local directory number for the B2 channel.
Step 15	`Router(config-if)# isdn caller number`	(Optional) Configure caller ID screening.
Step 16	`Router(config-if)# isdn answer1 [called-party-number][:subaddress]`	(Optional) Configures called-party number verification for a called-party number or subaddress number in the incoming setup message.
Step 17	`Router(config-if)# isdn calling-number calling-number`	(Optional) Specifies the calling-party number.
Step 18	`Router(config-if)# isdn not-end-to-end [56 | 64]`	(Optional) Configures the speed for incoming calls recognized as not ISDN end-to-end.
Step 19	`Router(config-if)# isdn incoming-voice modem`	Routes incoming voice calls to the modem and treats them as analog data. This step is required for the Modem over ISDN BRI feature.
Step 20	`Router(config-if)# isdn disconnect-cause {cause-code-number | busy | not available}`	Overrides specific cause codes such as modem availability and resource pooling that are sent to the switch by ISDN applications. When the **isdn disconnect-cause** command is implemented, the configured cause codes are sent to the switch; otherwise, the default cause codes of the application are sent. The *cause-code-number* argument sends a cause code number (submitted as integer 1 through 127) to the switch. The **busy** keyword sends the USER BUSY code to the switch. The **not available** keyword sends the CHANNEL NOT AVAILABLE code to the switch.
Step 21	`Router(config-if)# isdn fast-rollover-delay seconds`	(Optional) Configures a delay between fast rollover dials.
Step 22	`Router(config-if)# isdn sending-complete`	(Optional) Configures the BRI interface to include the Sending Complete information element in the outgoing call Setup message. 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.

Table 22 ISDN Switch Types

Country	ISDN Switch Type	Description
Australia	**basic-ts013**	Australian TS013 switches
Europe	**basic-1tr6**	German 1TR6 ISDN switches
	basic-net3	NET3 ISDN switches (United Kingdom and others)
	vn2	French VN2 ISDN switches
	vn3	French VN3 and VN4 ISDN switches
Japan	**ntt**	Japanese NTT ISDN switches
North America	**basic-5ess**	Lucent Technologies basic rate switches
	basic-dms100	NT DMS-100 basic rate switches
	basic-ni	National ISDN-1 switches

See the section "Configuration Examples for Modem over ISDN BRI" at the end of this chapter for configuration examples.

Verifying ISDN BRI Interface Configuration

Use the **show running-config** command in EXEC mode to verify the current configuration that is running on the terminal.

Note The **show startup-config** shows the configuration stored in NVRAM or in a location specified by the CONFIG_FILE environment variable.

The following example shows some of the command output that is relevant to BRI configuration tasks. The bold text in the example are the results of configuration steps such as those shown in the section "How to Configure Modem over ISDN BRI" earlier in this chapter.

```
Building configuration...

Current configuration:
!
version 12.0
no service udp-small-servers
service tcp-small-servers
!
hostname Router
!
enable secret 5 $1$c8xi$tObplXsIS.jDeo43yZgq50
enable password xxx
!
username xxxx password x 11x5xx07
no ip domain-lookup
ip host Labhost 172.17.12.1
ip host Labhost2 172.17.12.2
ip name-server 172.19.169.21
!
interface Ethernet0
 ip address 172.17.12.100 255.255.255.192
 no ip mroute-cache
```

```
 no ip route-cache
 no mop enabled
.
.
.
interface BRI1/7
 description (408) 555-3777
 ip address 10.1.1.26 255.255.255.1
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 no keepalive
 shutdown
 dialer idle-timeout 180
 dialer map ip 10.1.1.9 name MDial1 14085550715
 dialer map ip 10.1.1.14 name MDial2 14085553775
 dialer-group 1
 isdn switch-type basic-5ess
 isdn incoming-voice modem
 isdn disconnect-cause busy
no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink
.
.
.
!
interface Group-Async1
 ip unnumbered Loopback0
 no ip directed-broadcast
 ip tcp header-compression passive
 async mode interactive
 peer default ip address pool default
 no fair-queue
 group-range 65 70
 hold-queue 10 in
!
router igrp 109
 network 172.21.0.0
!
ip local pool local 172.21.50.85 172.21.50.89
ip local pool default 10.1.1.1 10.1.1.253
ip classless
ip route 0.0.0.0 0.0.0.0 172.21.48.1
!
!
map-class dialer VOICE
 dialer voice-call
!
map-class dialer DATA
dialer-list 1 protocol ip list 101
tacacs-server host 172.19.2.74
tacacs-server host 192.168.15.197
snmp-server community isdn RW
snmp-server enable traps isdn call-information
snmp-server host 172.25.3.154 traps isdn
```

Use the **show interfaces bri** *number* command to verify information about the physical attributes of the ISDN BRI B and D channels. The *number* argument is the slot location of the BRI module. Valid values are from 0 to 3.

```
BRI0:1 is down, line protocol is down
 Hardware is BRI
 MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
 Internet address is 10.1.1.3/27
 Encapsulation PPP, loopback not set, keepalive not set
 LCP Closed
 Closed: IPCP
 Last input never, output never, output hang never
 Last clearing of "show interface" counters never
 Queueing strategy: fifo
 Output queue 0/40, 0 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
 0 packets input, 0 bytes, 0 no buffer
 Received 0 broadcasts, 0 runts, 0 giants
 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, 7 interface resets
 0 output buffer failures, 0 output buffers swapped out
 0 carrier transitions
```

Configuration Examples for Modem over ISDN BRI

This section provides the following examples:

- BRI Interface Configuration Example
- Complete Configuration Examples

These examples show configuration of just the Modem over ISDN BRI feature using the interface configuration commands for each interface and a complete configuration showing global configuration, BRI interfaces, and modem configuration.

BRI Interface Configuration Example

The following example shows how to configure each BRI interface on a Cisco 3640 router for the Modem over ISDN BRI feature:

```
interface BRI0/0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000101 9194440001
 isdn spid2 0444001101 9194440011
 isdn incoming-voice modem
!
interface BRI0/1
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000201 9194440002
 isdn spid2 0444001201 9194440012
 isdn incoming-voice modem
!
interface BRI0/2
 no ip address
 no ip directed-broadcast
```

```
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000301 9194440003
 isdn spid2 0444001301 9194440013
 isdn incoming-voice modem
!
interface BRI0/3
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000401 9194440004
 isdn spid2 0444001401 9194440014
 isdn incoming-voice modem
!
interface BRI0/4
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000501 9194440005
 isdn spid2 0444001501 9194440015
 isdn incoming-voice modem
!
interface BRI0/5
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000601 9194440006
 isdn spid2 0444001601 9194440016
 isdn incoming-voice modem
!
interface BRI0/6
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000701 9194440007
 isdn spid2 0444001701 9194440017
 isdn incoming-voice modem
!
interface BRI0/7
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000801 9194440008
 isdn spid2 0444001801 9194440018
 isdn incoming-voice modem
!
interface BRI2/0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000101 9195550001
 isdn spid2 0555001101 9195550011
 isdn incoming-voice modem
!
interface BRI2/1
 no ip address
 no ip directed-broadcast
 encapsulation ppp
```

```
 isdn switch-type basic-ni
 isdn spid1 0555000201 9195550002
 isdn spid2 0555001201 9195550012
 isdn incoming-voice modem
!
interface BRI2/2
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000301 9195550003
 isdn spid2 0555001301 9195550013
 isdn incoming-voice modem
!
interface BRI2/3
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000401 9195550004
 isdn spid2 0555001401 9195550014
 isdn incoming-voice modem
!
interface BRI2/4
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000501 9195550005
 isdn spid2 0555001501 9195550015
 isdn incoming-voice modem
!
interface BRI2/5
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000601 9195550006
 isdn spid2 0555001601 9195550016
 isdn incoming-voice modem
!
interface BRI2/6
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000701 9195550007
 isdn spid2 0555001701 9195550017
 isdn incoming-voice modem
!
interface BRI2/7
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000801 9195550008
 isdn spid2 0555001801 9195550018
 isdn incoming-voice modem
!
```

Complete Configuration Examples

The following example shows a complete configuration for a dial-in router, including a global command, BRI interface configuration, and modem configuration including **group-async** and **dialer** commands.

```
version 12.0
service timestamps debug datetime localtime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname MBRI_IN
!
no logging buffered
enable password xxx
```

The following lines are used for PPP CHAP authentication. Each username and password is associated with one dialer interface.

```
username async1 password devtest
username async2 password devtest
username async3 password devtest
username async4 password devtest
username async5 password devtest
username async6 password devtest
username async7 password devtest
username async8 password devtest
username async9 password devtest
username async10 password devtest
username async11 password devtest
username async12 password devtest
username async13 password devtest
username async14 password devtest
username async15 password devtest
username async16 password devtest
username async17 password devtest
username async18 password devtest
username async19 password devtest
username async20 password devtest
username async21 password devtest
username async22 password devtest
username async23 password devtest
username async24 password devtest
username async25 password devtest
username async26 password devtest
username async27 password devtest
username async28 password devtest
username async29 password devtest
username async30 password devtest
username FLOYD password devtest
username MBRI_OUT password devtest
ip subnet-zero
no ip domain-lookup
!
isdn switch-type basic-5ess
```

```
interface BRI0/0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000101 9194440001
 isdn spid2 0444001101 9194440011
 isdn incoming-voice modem
!
interface BRI0/1
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000201 9194440002
 isdn spid2 0444001201 9194440012
 isdn incoming-voice modem
!
interface BRI0/2
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000301 9194440003
 isdn spid2 0444001301 9194440013
 isdn incoming-voice modem
!
interface BRI0/3
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000401 9194440004
 isdn spid2 0444001401 9194440014
 isdn incoming-voice modem
!
interface BRI0/4
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000501 9194440005
 isdn spid2 0444001501 9194440015
 isdn incoming-voice modem
 no shut
!
interface BRI0/5
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000601 9194440006
 isdn spid2 0444001601 9194440016
 isdn incoming-voice modem
!
interface BRI0/6
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000701 9194440007
 isdn spid2 0444001701 9194440017
 isdn incoming-voice modem
!
```

```
interface BRI0/7
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0444000801 9194440008
 isdn spid2 0444001801 9194440018
 isdn incoming-voice modem
!
interface BRI2/0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000101 9195550001
 isdn spid2 0555001101 9195550011
 isdn incoming-voice modem
!
interface BRI2/1
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000201 9195550002
 isdn spid2 0555001201 9195550012
 isdn incoming-voice modem
!
interface BRI2/2
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000301 9195550003
 isdn spid2 0555001301 9195550013
 isdn incoming-voice modem
!
interface BRI2/3
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000401 9195550004
 isdn spid2 0555001401 9195550014
 isdn incoming-voice modem
!
interface BRI2/4
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000501 9195550005
 isdn spid2 0555001501 9195550015
 isdn incoming-voice modem
!
interface BRI2/5
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000601 9195550006
 isdn spid2 0555001601 9195550016
 isdn incoming-voice modem
!
```

```
interface BRI2/6
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000701 9195550007
 isdn spid2 0555001701 9195550017
 isdn incoming-voice modem
!
interface BRI2/7
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 isdn switch-type basic-ni
 isdn spid1 0555000801 9195550008
 isdn spid2 0555001801 9195550018
 isdn incoming-voice modem
!
interface Ethernet1/0
 ip address 172.18.16.123 255.255.255.192
 no ip directed-broadcast
!
```

The following example defines a group-async interface for grouping all the digital modems and configuring them together. Group-async configuration is much easier than configuring all 30 digital modems individually.

```
interface Group-Async1
 ip unnumbered Ethernet3/1
 no ip directed-broadcast
 encapsulation ppp
 load-interval 30
 dialer in-band
 dialer pool-member 1
 async default routing
 async mode dedicated
 no peer default ip address
 no cdp enable
 ppp authentication chap
 group-range 96 125
 hold-queue 10 in
```

The following example defines dialer interfaces, associates IP addresses, and sets all the authentication parameters required during the call establishment.

```
interface Dialer1
 ip address 10.1.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async1
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async1
 ppp chap password devtest
!
interface Dialer2
 ip address 10.2.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async2
 dialer pool 1
 dialer-group 1
 no cdp enable
```

```
 ppp authentication chap callin
 ppp chap hostname async2
 ppp chap password devtest
!
interface Dialer3
 ip address 10.3.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async3
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async3
 ppp chap password devtest
!
interface Dialer4
 ip address 10.4.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async4
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async4
 ppp chap password devtest
!
interface Dialer5
 ip address 10.5.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async5
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async5
 ppp chap password devtest
!
interface Dialer6
 ip address 10.6.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async6
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async6
 ppp chap password devtest
!
interface Dialer7
 ip address 10.7.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async7
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async7
 ppp chap password devtest
!
```

```
interface Dialer8
 ip address 10.8.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async8
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async8
 ppp chap password devtest
!
interface Dialer9
 ip address 10.9.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async9
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async9
 ppp chap password devtest
!
interface Dialer10
 ip address 10.10.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async10
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async10
 ppp chap password devtest
!
interface Dialer11
 ip address 10.11.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async11
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async11
 ppp chap password devtest
!
interface Dialer12
 ip address 10.12.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async12
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async12
 ppp chap password devtest
!
interface Dialer13
 ip address 10.13.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
```

```
 dialer remote-name async13
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async13
 ppp chap password devtest
!
interface Dialer14
 ip address 10.14.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async14
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async14
 ppp chap password devtest
!
interface Dialer15
 ip address 10.15.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async15
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async15
 ppp chap password devtest
!
interface Dialer16
 ip address 10.16.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async16
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async16
 ppp chap password devtest
!
interface Dialer17
 ip address 10.17.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async17
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async17
 ppp chap password devtest
!
interface Dialer18
 ip address 10.18.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async18
 dialer pool 1
 dialer-group 1
 no cdp enable
```

```
 ppp authentication chap callin
 ppp chap hostname async18
 ppp chap password devtest
!
interface Dialer19
 ip address 10.19.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async19
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async19
 ppp chap password devtest
!
interface Dialer20
 ip address 10.20.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async20
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async20
 ppp chap password devtest
!
interface Dialer21
 ip address 10.21.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async21
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async21
 ppp chap password devtest
!
interface Dialer22
 ip address 10.22.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async22
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async22
 ppp chap password devtest
!
interface Dialer23
 ip address 10.23.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async23
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async23
 ppp chap password devtest
!
```

```
interface Dialer24
 ip address 10.24.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async24
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async24
 ppp chap password devtest
!
interface Dialer25
 ip address 10.25.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async25
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async25
 ppp chap password devtest
!
interface Dialer26
 ip address 10.26.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async26
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async26
 ppp chap password devtest
!
interface Dialer27
 ip address 10.27.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async27
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async27
 ppp chap password devtest
!
interface Dialer28
 ip address 10.28.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async28
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async28
 ppp chap password devtest
!
interface Dialer29
 ip address 10.29.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
```

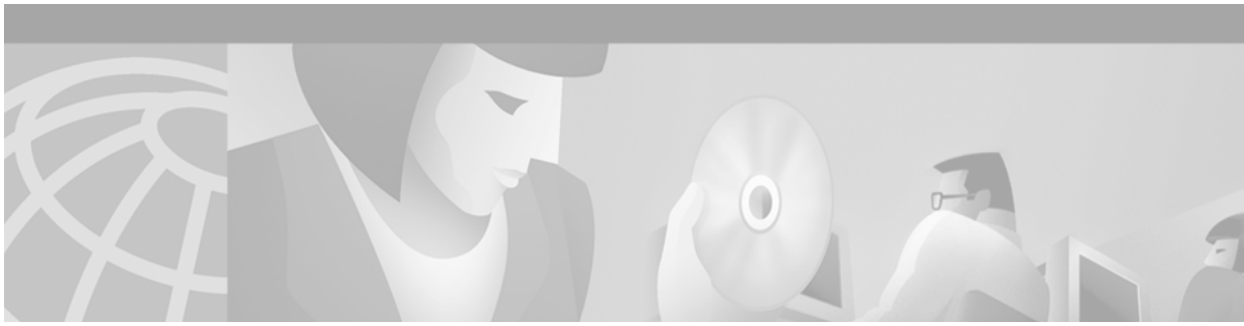
```
 dialer remote-name async29
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async29
 ppp chap password devtest
!
interface Dialer30
 ip address 10.30.0.1 255.255.0.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name async30
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap callin
 ppp chap hostname async30
 ppp chap password devtest
!
no ip classless
```

The following lines define routes that send incoming packets out via specific interfaces:

```
ip route 0.0.0.0 0.0.0.0 172.18.16.193
ip route 10.91.0.1 255.255.255.255 1.1.0.2
ip route 10.91.0.2 255.255.255.255 1.2.0.2
ip route 10.91.0.3 255.255.255.255 1.3.0.2
ip route 10.91.0.4 255.255.255.255 1.4.0.2
ip route 10.91.0.5 255.255.255.255 1.5.0.2
ip route 10.91.0.6 255.255.255.255 1.6.0.2
ip route 10.91.0.7 255.255.255.255 1.7.0.2
ip route 10.91.0.8 255.255.255.255 1.8.0.2
ip route 10.91.0.9 255.255.255.255 1.9.0.2
ip route 10.91.0.10 255.255.255.255 1.10.0.2
ip route 10.91.0.11 255.255.255.255 1.11.0.2
ip route 10.91.0.12 255.255.255.255 1.12.0.2
ip route 10.91.0.13 255.255.255.255 1.13.0.2
ip route 10.91.0.14 255.255.255.255 1.14.0.2
ip route 10.91.0.15 255.255.255.255 1.15.0.2
ip route 10.91.0.16 255.255.255.255 1.16.0.2
ip route 10.91.0.17 255.255.255.255 1.17.0.2
ip route 10.91.0.18 255.255.255.255 1.18.0.2
ip route 10.91.0.19 255.255.255.255 1.19.0.2
ip route 10.91.0.20 255.255.255.255 1.20.0.2
ip route 10.91.0.21 255.255.255.255 1.21.0.2
ip route 10.91.0.22 255.255.255.255 1.22.0.2
ip route 10.91.0.23 255.255.255.255 1.23.0.2
ip route 10.91.0.24 255.255.255.255 1.24.0.2
ip route 10.91.0.25 255.255.255.255 1.25.0.2
ip route 10.91.0.26 255.255.255.255 1.26.0.2
ip route 10.91.0.27 255.255.255.255 1.27.0.2
ip route 10.91.0.28 255.255.255.255 1.28.0.2
ip route 10.91.0.29 255.255.255.255 1.29.0.2
ip route 10.91.0.30 255.255.255.255 1.30.0.2
ip route 172.18.0.0 255.255.0.0 Ethernet3/1
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
 transport input none
```

The following example configures the lines associated with the digital modems:

```
line 96 125
 exec-timeout 0 0
 modem InOut
 transport input all
 stopbits 1
 flowcontrol hardware
line aux 0
 exec-timeout 0 0
line vty 0 4
 exec-timeout 0 0
 password lab
 login
line vty 5 60
 exec-timeout 0 0
 password lab
 login
!
end
```



Configuring X.25 on ISDN

This chapter describes how to configure X.25 on ISDN. It includes the following main sections:

- X.25 on ISDN Overview
- How to Configure X.25 on ISDN
- Configuration Examples for X.25 on ISDN

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

X.25 on ISDN Overview

BRI is an ISDN interface, and it consists of two B channels (B1 and B2) and one D channel. The B channels are used to transfer data, voice, and video. The D channel controls the B channels.

ISDN uses the D channel to carry signal information. ISDN can also use the D channel in a BRI to carry X.25 packets. The D channel has a capacity of 16 kbps, and the X.25 over D channel can utilize up to 9.6 kbps.

X.25-over-D-Channel Logical Interface

When X.25 on ISDN is configured, a separate X.25-over-D-channel logical interface is created. You can set its parameters without disrupting the original ISDN interface configuration. The original BRI interface will continue to represent the D, B1, and B2 channels.

Because some end-user equipment uses static terminal endpoint identifiers (TEIs) to access this feature, static TEIs are supported. The dialer understands the X.25-over-D-channel calls and initiates them on a new interface.

X.25 traffic over the D channel can be used as a primary interface where low-volume, sporadic interactive traffic is the normal mode of operation. Supported traffic includes the Internet Protocol Exchange (IPX), AppleTalk, transparent bridging, Xerox Network Systems (XNS), DECnet, and IP.

This feature is not available on the ISDN PRI.

Note X.25 on ISDN is also supported using the ISDN Always On/Dynamic (AO/DI) feature. AO/DI uses the Multilink PPP (MLP) protocol signaling with standard Q.922 and X.25 encapsulations, and can additionally use the Bandwidth Allocation Control Protocol (BACP) to optimize bandwidth on demand. For information about how to configure AO/DI, see the chapter "Configuring X.25 on ISDN Using AO/DI" in this publication.

Outbound Circuit-Switched X.25 Support over a Dialer Interface

Current Cisco IOS software enablescircuit-switched X.25 clients—PAD, X.25 switching, and Qualified Logical Link Control (QLLC)—to initiate calls and dynamically bring the X.25 context (which runs the X.25 protocol) up or down as needed. This capability allows packet-switched traffic over ISDN.

In earlier releases of the Cisco IOS software, X.25 circuit-switched clients were required to do an X.25 route lookup to forward a call. If the lookup resulted in a route to a dialer interface, the client would check the X.25 protocol state on the dialer interface. If the interface was not already bound to run the X.25 protocol, the software would reroute the call instead of bringing up a link and running the X.25 protocol. With this new feature, the X.25 context is dynamically created on demand and then removed when the X.25 session is cleared on the dialer interface.

For dialer profile interfaces, the X.25 context is created on the dialer interface, because X.25 protocol functions run on the dialer interface itself. Member links act like forwarding devices, because their topmost interface runs the actual encapsulated protocol. But for legacy dialer interfaces, the X.25 context is created on the member links once they come up and bind to a dialer.

There are no specific configuration tasks required to enable outbound circuit-switched X.25 support. See the "Outbound Circuit-Switched X.25 Example" example in the section "Configuration Examples for X.25 on ISDN" at the end of this chapter for an example of how to make use of this feature in your network.

How to Configure X.25 on ISDN

You can configure X.25 on ISDN in three ways:

- If the ISDN traffic will cross an X.25 network, you configure the ISDN interface as described in the "Setting Up Basic ISDN Services" and "Configuring signaling on T1 and E1" chapters earlier in this publication. Make certain to configure that ISDN interface for X.25 addressing and encapsulation as described in the "Configuring X.25" chapter of the *Cisco IOS Wide-Area Networking Configuration Guide*.
- Configure dynamic X.25 as illustrated in the section "Outbound Circuit-Switched X.25 Example" later in this chapter.
- If the D channel of an ISDN BRI interface is to carry X.25 traffic, perform the task described in the next section, "Configuring X.25 on the ISDN D Channel."

Configuring X.25 on the ISDN D Channel

To configure an ISDN BRI interface (and create a special ISDN interface) to carry X.25 traffic on the D channel, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri number`	Specifies an ISDN BRI interface and begins interface configuration mode.
Step 2	`Router(config-if)# isdn x25 static-tei tei-number`	Specifies a static TEI, if required by the switch.
Step 3	`Router(config-if)# isdn x25 dchannel`	Creates a configurable interface for X.25 traffic over the ISDN D channel.

The last step is to configure the X.25-over-ISDN interface for X.25 traffic. See the chapter "Configuring LAPB and X.25" in the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2, for the commands and tasks.

The new X.25-over-ISDN interface is called **interface bri** *number***:0** in configuration displays. It must be configured as an individual X.25 interface. For information about configuring an interface for X.25 traffic, refer to the *Cisco IOS Wide-Area Networking Configuration Guide*, Release 12.2.

Note The **encapsulation x25** command is neither required nor used on this new interface, but other X.25 commands can be used to configure this interface.

If you want to remove the X.25-over-ISDN interface later, use the **no isdn x25 dchannel** command.

See the section "X.25 on ISDN D-Channel Configuration Example" at the end of this chapter for a configuration example.

Configuration Examples for X.25 on ISDN

This section illustrates X.25 on ISDN with the following examples:

- X.25 on ISDN D-Channel Configuration Example
- Outbound Circuit-Switched X.25 Example

X.25 on ISDN D-Channel Configuration Example

The following example creates a BRI 0:0 interface for X.25 traffic over the D channel and then configures the new interface to carry X.25 traffic:

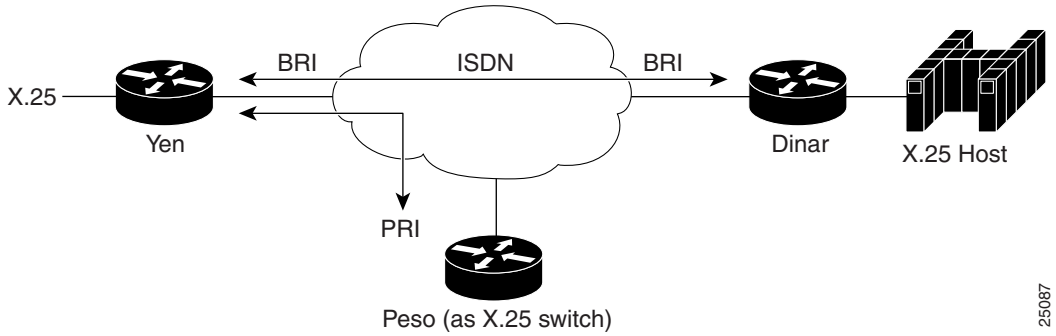
```
interface bri0
 isdn x25 dchannel
 isdn x25 static-tei 8
!
interface bri0:0
 ip address 10.1.1.2 255.255.255.0
 x25 address 31107000000100
 x25 htc 1
 x25 suppress-calling-address
```

```
x25 facility windowsize 2 2
x25 facility packetsize 256 256
x25 facility throughput 9600 9600
x25 map ip 10.1.1.3 31107000000200
```

Outbound Circuit-Switched X.25 Example

The following example shows how to configure dynamic X.25 on an ISDN interface. Figure 33 illustrates the configuration.

Figure 33 Dynamic X.25 over ISDN



Configuration for Yen

```
version 12.0(5)T
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname yen
!
enable secret 5 $1$K32j$4AZW2oMDivpUeuMa/Fdcd.
enable password secret
!
username peso password 0 cisco
username dinar password 0 cisco
ip subnet-zero
no ip domain-lookup
ip domain-name cicso.com
ip name-server 172.18.1.148
!
isdn switch-type basic-5ess
x25 routing
!
interface Loopback0
no ip address
no ip directed-broadcast
no ip mroute-cache
!
interface Ethernet0
ip address 172.21.75.2 255.255.255.0
no ip directed-broadcast
no ip mroute-cache
media-type 10BaseT
!
```

```
interface BRI1
no ip address
no ip directed-broadcast
no ip mroute-cache
dialer pool-member 1
isdn switch-type basic-5ess
no fair-queue
!
interface Dialer0
ip address 10.1.1.1 255.0.0.0
no ip directed-broadcast
encapsulation x25
no ip mroute-cache
dialer remote-name dinar
dialer idle-timeout 180
dialer string 81060
dialer caller 81060
dialer max-call 1
dialer pool 1
dialer-group 1
x25 address 11111
x25 map ip 10.1.1.2 22222
!
ip default-gateway 172.21.75.1
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.75.1
no ip http server
!
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!
x25 route 22222 interface Dialer0
x25 route 33333 interface Dialer0
!
line con 0
exec-timeout 0 0
transport input none
line aux 0
transport input all
line vty 0 4
password cisco
login
line vty 5 100
password cisco
login
!
end
```

Configuration for Peso Acting as X.25 Switch

```
version 12.0(5)T
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname peso
!
enable secret 5 $1$.Q00$h3vIhbOwO1fPvA2LYx2gE.
enable password cisco
!
ip subnet-zero
!
isdn switch-type primary-5ess
x25 routing
```

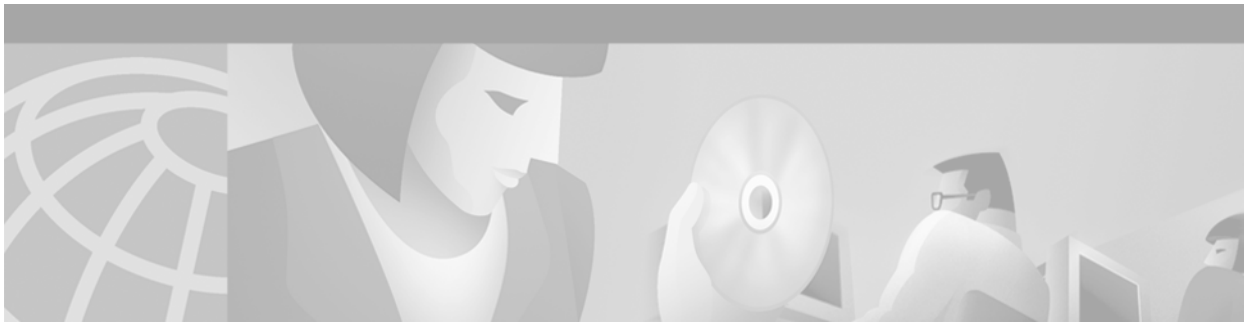
```
!
controller T1 0
cablelength short
cablelength short 133
!
controller T1 1
framing esf
clock source line primary
pri-group timeslots 1-24
!
controller T1 2
cablelength short
cablelength short 133
!
controller T1 3
cablelength short
cablelength short 133
!
interface Ethernet0
ip address 172.21.75.3 255.255.255.0
no ip directed-broadcast
!
interface Serial1:23
no ip address
no ip directed-broadcast
encapsulation ppp
dialer pool-member 1
isdn switch-type primary-5ess
isdn incoming-voice modem
no fair-queue
no cdp enable
ppp authentication chap
!
interface Dialer0
no ip address
no ip directed-broadcast
encapsulation x25 dce
no ip mroute-cache
dialer remote-name yen
dialer idle-timeout 180
dialer string 61401
dialer caller 61401
dialer max-call 1
dialer pool 1
x25 address 33333
!
interface Dialer1
no ip address
no ip directed-broadcast
encapsulation x25 dce
no ip mroute-cache
dialer remote-name dinar
dialer idle-timeout 180
dialer string 61403
dialer caller 61403
dialer max-call 1
dialer pool 1
x25 address 44444
!
ip default-gateway 172.21.75.1
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.75.1
no ip http server
!
```

```
x25 route 11111 interface Dialer0
x25 route 22222 interface Dialer1
x25 route source 11111 interface Dialer1
x25 route input-interface Dialer0 interface Dialer1
!
line con 0
transport input none
line 1 48
line aux 0
line vty 0 4
password cisco
login
line vty 5 100
password cisco
login
!
end
```

Configuration for Dinar

```
version 12.0(5)T
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname dinar
!
logging buffered 16384 debugging
enable secret 5 $1$8EjF$4.S0AoMOVa5OIAYEMrrFI/
enable password cisco
!
username yen password 0 cisco
username 7701
username drachma password 0 cisco
username AODI password 0 cisco
ip subnet-zero
ip rcmd rcp-enable
ip rcmd rsh-enable
ip rcmd remote-username atirumal
!
isdn switch-type basic-5ess
x25 routing
!
controller T1 0/0
!
interface BRI3/1
no ip address
no ip directed-broadcast
no ip mroute-cache
dialer pool-member 1
isdn switch-type basic-5ess
no fair-queue
!
interface Dialer0
ip address 10.1.1.2 255.0.0.0
no ip directed-broadcast
encapsulation x25
no ip mroute-cache
dialer remote-name yen
dialer idle-timeout 180
dialer string 81060
dialer caller 81060
dialer max-call 1
dialer pool 1
```

```
dialer-group 1
x25 address 22222
x25 map ip 10.1.1.1 11111
!
interface Dialer1
ip address 10.1.1.10 255.0.0.0
no ip directed-broadcast
no ip mroute-cache
dialer in-band
dialer-group 1
no fair-queue
!
ip default-gateway 172.21.75.1
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.75.1
no ip http server
!
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!
x25 route 11111 interface Dialer0
x25 route 44444 interface Dialer0
!
```



Configuring X.25 on ISDN Using AO/DI

The chapter describes how to configure the X.25 on ISDN using the Always On/Dynamic ISDN (AO/DI) feature. It includes the following main sections:

- AO/DI Overview
- How to Configure an AO/DI Interface
- How to Configure an AO/DI Client/Server
- Configuration Examples for AO/DI

AO/DI supports PPP encapsulation on switched X.25 virtual circuits (VCs) only.

The X.25 encapsulation (per RFC 1356), PPP, Bandwidth Allocation Control Protocol (BACP), and Bandwidth Allocation Protocol (BAP) modules must be present in both the AO/DI client and server.

AO/DI relies on features from X.25, PPP, and BACP modules and must be configured on both the AO/DI client and server. BAP, if negotiated, is a subset of BACP, which is responsible for bandwidth allocation for the Multilink PPP (MLP) peers. It is recommended you configure MLP with the BAP option due to the differences between the ISDN (E.164) and X.25 (X.121) numbering formats.

To implement AO/DI, you must configure the AO/DI client and server for PPP, incorporating BAP and X.25 module commands. This task involves configuring the BRI or PRI interfaces with the appropriate X.25 commands and the dialer interfaces with the necessary PPP or BAP commands.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

AO/DI Overview

AO/DI functionality is based on the technology modules described in the following sections:

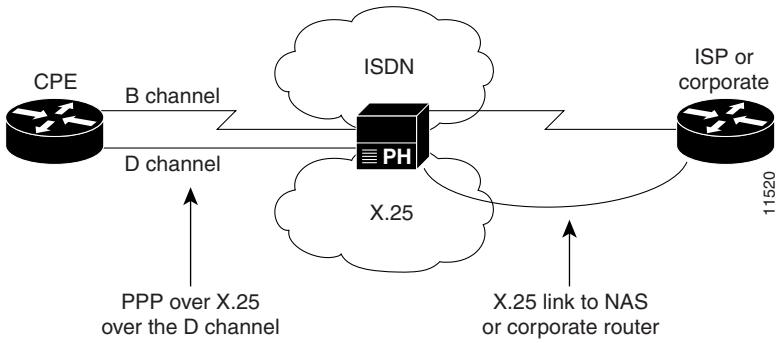
- PPP over X.25 Encapsulation
- Multilink PPP Bundle
- BACP/BAP

AO/DI is an on-demand service that is designed to optimize the use of an existing ISDN signaling channel (D channel) to transport X.25 traffic. The X.25 D-channel call is placed from the subscriber to the packet data service provider. The use of PPP allows protocols to be encapsulated within the X.25 logical circuit carried by the D channel. The bearer channels (B channels) use the multilink protocol without the standard Q.922 and X.25 encapsulations, and invoke additional bandwidth as needed. Optionally, BACP and BAP can be used to negotiate bandwidth allocation as required.

AO/DI takes full advantage of existing packet handlers at the central office by using an existing D channel to transport the X.25 traffic. The link associated with the X.25 D channel packet connection is used as the primary link of the multilink bundle. The D channel is a connectionless, packet-oriented link between the customer premise equipment (CPE) and the central office. Because the D channel is always available, it is possible to in turn offer "always available" services. On-demand functionality is achieved by using the B channels to temporarily boost data throughput and by disconnecting them after use. Figure 34 shows the AO/DI environment and how ISDN and X.25 resources are implemented.

Note On the client side, the X.25 switched virtual circuit (SVC) can only be terminated on an ISDN D channel; however, on the server side, the SVC can be terminated on an ISDN BRI using a D channel, a PRI using specific time slots, or a high-speed serial link.

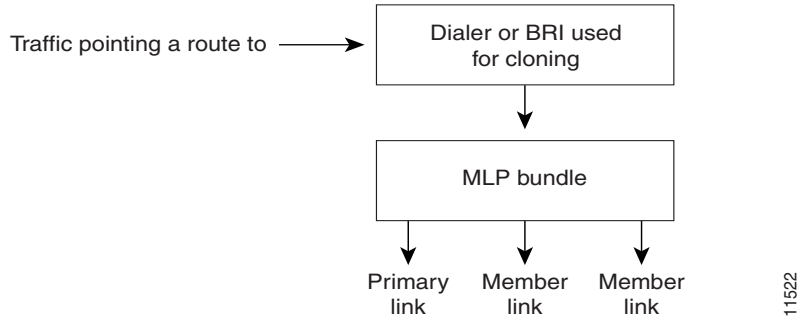
Figure 34 AO/DI Environment



AO/DI provides the following benefits:

- ISDN telecommuting cost savings. Low-speed, D-channel services are typically more cost-efficient than the time-based tariffs applied to the B channels, which usually carry user data.
- Reductions in the amount of data traffic from service provider voice networks. The D-channel X.25 packets are handled at the central office by the X.25 packet handler, thereby routing these packets bypassing the switch, which reduces impact on the telephony network.
- Network access server cost reductions. AO/DI can reduce service provider network access server costs by increasing port efficiencies. Initial use of the "always on" D-channel connection lowers the contention ratio on standard circuit switched dial ports. (See Figure 35.)

Figure 35 Increasing Port Efficiency with AO/DI

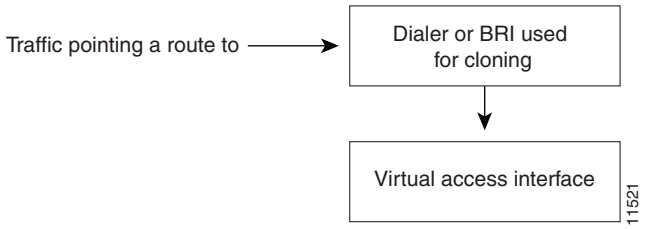


PPP over X.25 Encapsulation

PPP over X.25 is accomplished through the following process:

1. The X.25 map statement on the client side creates a virtual access interface. A virtual access interface is dynamically created and configured by cloning the configuration from a dialer interface (dialer interface 1, for example).
2. The dialer interface goes into "spoofing" mode and stays in this mode until interesting traffic is seen.
3. When interesting traffic is seen, the dialer interface activates the virtual access interface, which creates the X.25 SVC. Once the SVC is established, PPP negotiation begins in order to bring up the line protocol. The client will initiate a call to the remote end server, per the **x25 map ppp** command.
4. When the AO/DI server receives a call intended for its X.25 map statement, the call is accepted and an event is queued to the X.25 encapsulation manager. The encapsulation manager is an X.25 process that authenticates incoming X.25 calls and AO/DI events, and creates a virtual access interface that clones the configuration from the dialer or BRI interface. Figure 36 shows the virtual interface creation process.

Figure 36 Creating a Virtual Access Interface

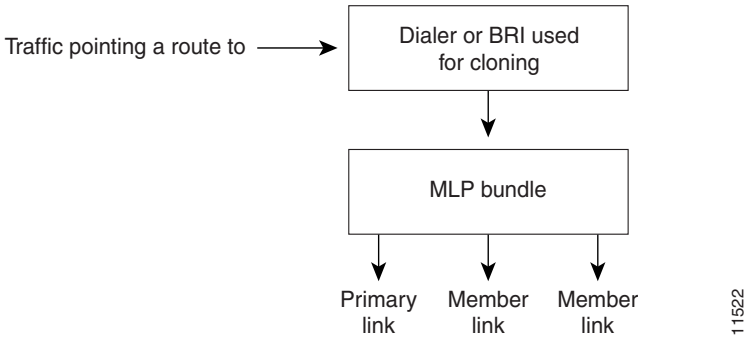


Multilink PPP Bundle

The multilink protocol offers load balancing, packet fragmentation, and the bandwidth allocation functionality that is key to AO/DI structure. The MLP bundle process is achieved through the following process:

1. The **ppp multilink bap** command initiates MLP and, subsequently, BAP. The virtual access interface that is created above the X.25 VC (over the D channel) becomes the first member link of the MLP bundle.
2. The **ppp multilink idle-link** command works in conjunction with the **dialer load-threshold** command in order to add B channels as needed to boost traffic throughput. When a B channel is added, the first member link enters "receive only" mode, allowing the link additions. When the higher throughput is no longer needed, the additional B channels are disconnected and the primary link is the only link in the bundle, the bundle disengages "receive only" mode. The X.25 SVC stays active. Figure 37 shows the MLP bundle sequence.

Figure 37 MLP Bundle Creation Sequence



MLP Encapsulation Enhancements

In previous releases of the Cisco IOS software, when MLP was used in a dialer profile, a virtual access interface was always created as the bundle. It was bound to both the B channel and the dialer profile interfaces after creation and cloning. The dialer profile interface could act as the bundle without help from a virtual access interface. But with recent software enhancements, it is no longer the virtual access interface that is added into the connected group of the dialer profile, but the dialer profile itself. The dialer profile becomes a connected member of its own connected group.

BACP/BAP

Bandwidth resources are provided by BACP, described in RFC 2125. Once the MLP peers have successfully negotiated BACP, BAP negotiates bandwidth resources in order to support traffic throughput. BAP is a subset of BACP, and it defines the methods and governing rules for adding and removing links from the bundle for MLP. BACP/BAP negotiations are achieved through the following process:

1. Once the MLP session is initiated and BACP is negotiated over the MLP bundle, the AO/DI client issues a BAP call request for additional bandwidth.
2. The AO/DI server responds with the BAP call response, which contains the phone number of the B channel to add. B channels are added, as needed, to support the demand for increased traffic throughput.
3. B channels are disconnected as the traffic load decreases.

How to Configure an AO/DI Interface

To configure X.25 on ISDN using AO/DI, perform the following tasks:

- Configuring PPP and BAP on the Client (As required)
- Configuring X.25 Parameters on the Client (As required)
- Configuring PPP and BAP on the Server (As required)
- Configuring X.25 Parameters on the Server (As required)

For examples of how to configure X.25 on ISDN using AO/DI in your network, see the section "Configuration Examples for AO/DI" at the end of this chapter.

Configuring PPP and BAP on the Client

To configure PPP and BAP under the dialer interface on the AO/DI client, use the following commands in interface configuration mode as needed:

Command	Purpose
`Router(config-if)# ppp multilink bap`	Enables PPP BACP bandwidth allocation negotiation.
`Router(config-if)# encapsulation ppp`	Enables PPP on the interface.
`Router(config-if)# dialer in-band`	Enables dial-on-demand routing (DDR) on the interface.
`Router(config-if)# dialer load-threshold load`	Sets the dialer load threshold.
`Router(config-if)# dialer-group group-number`	Controls access to this interface by adding it to a dialer access group.
`Router(config-if)# ppp bap callback accept`	(Optional) Enables the interface to initiate additional links upon peer request.

Command	Purpose
`Router(config-if)# ppp bap call request`	Enables the interface to initiate additional links.

`Router(config-if)# dialer map` *protocol next-hop-address* `[name` *hostname*`] [spc] [speed 56 | speed 64] [broadcast] [modem-script` *modem-regexp*`] system-script` *system-regexp*`]`
 	
or
 	
`Router(config-if)# dialer string` *dial-string*`[:`*isdn-subaddress*`]`
 	
`Router(config-if)# dialer string` *dial-string* `[class` *class-name*`]`
	Enables a serial interface or an ISDN interface to initiate and receive calls to or from remote sites.
	
	Specifies the destination string (telephone number) for calling:
	• A single site (using legacy DDR)
	• Multiple sites (using dialer profiles)

Configuring X.25 Parameters on the Client

The AO/DI client interface must be configured to run PPP over X.25. To configure the interface for the X.25 parameters, use the following commands in interface configuration mode as needed:

Command	Purpose
`Router(config-if)# x25 address` *address*	Configures the X.25 address.
`Router(config-if)# x25 htc` *circuit-number*	Sets the highest two-way circuit number. For X.25 the default is 1024.
`Router(config-if)# x25 win` *packets*	Sets the default VC receive window size. The default is 2 packets.[1]
`Router(config-if)# x25 wout` *packets*	Sets the default VC transmit window size. The default is 2 packets.[1]

1. The default input and output window sizes are typically defined by your network administrator. Cisco IOS configured window sizes must be set to match the window size of the network.

For details and usage guidelines for X.25 configuration parameters, refer to the *Cisco IOS Wide-Area Networking Configuration Guide* and *Cisco IOS Wide-Area Networking Command Reference*.

Configuring PPP and BAP on the Server

To configure PPP and BAP under the dialer interface on the AO/DI server, use the following commands in interface configuration mode as needed:

Command	Purpose
`Router(config-if)# ppp multilink bap`	Enables PPP BACP bandwidth allocation negotiation.
`Router(config-if)# encapsulation ppp`	Enables PPP on the interface.
`Router(config-if)# dialer in-band`	Enables DDR on the interface.

Command	Purpose
`Router(config-if)# dialer load-threshold load`	Sets the dialer load threshold.
`Router(config-if)# dialer-group group-number`	Controls access to this interface by adding it to a dialer access group.
`Router(config-if)# ppp bap call accept`	Enables the interface to accept additional links upon peer request.
`Router(config-if)# ppp bap callback request`	Enables the interface to initiate additional links (optional).

BAP configuration commands are optional. For information on how to configure BACP/BAP see the chapter "Configuring BACP" later in this publication.

Configuring X.25 Parameters on the Server

The AO/DI server BRI, PRI, or serial interface must be configured for the X.25 parameters necessary to run PPP over X.25. To configure the interface for X.25 parameters, use the following commands in interface configuration mode as needed:

Command	Purpose
`Router(config-if)# x25 address address`	Configures the X.25 address.
`Router(config-if)# x25 htc circuit-number`	Sets the highest two-way circuit number. For X.25 the default is 1024.
`Router(config-if)# x25 win packets`	Sets the default VC receive window size. The default is 2 packets.[1]
`Router(config-if)# x25 wout packets`	Sets the default VC transmit window size. The default is 2 packets.[1]

1. The default input and output window sizes are typically defined by your network administrator. Cisco IOS configured window sizes must be sets to match the window size of the network.

For details and usage guidelines for X.25 configuration parameters, see the *Cisco IOS Wide-Area Networking Configuration Guide* and *Cisco IOS Wide-Area Networking Command Reference.*

How to Configure an AO/DI Client/Server

Once the AO/DI client and server are configured with the necessary PPP, BAP, and X.25 commands, configure the routers to perform AO/DI. Perform the tasks in the following sections:

- Configuring the AO/DI Client (Required)
- Configuring the AO/DI Server (Required)

Configuring the AO/DI Client

To configure AO/DI, you must complete the tasks in the following section. The last task, to define local number peer characteristics, is optional.

- Enabling AO/DI on the Interface (Required)
- Enabling the AO/DI Interface to Initiate Client Calls (Required)
- Enabling the MLP Bundle to Add Multiple Links (Required)
- Modifying BACP Default Settings (Optional)

See the section "AO/DI Client Configuration Example" at the end of this chapter for an example of how to configure the AO/DI client.

Enabling AO/DI on the Interface

To enable an interface to run the AO/DI client, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# x25 aodi`	Enables the AO/DI client on an interface.

Enabling the AO/DI Interface to Initiate Client Calls

You must enable the interface to establish a PPP session over the X.25 protocol. The cloning interface will hold the PPP configuration, which will be cloned by the virtual access interface that is created and attached to the X.25 VC. The cloning interface must also hold the MLP configuration that is needed to run AO/DI.

To add the X.25 map statement that will enable the PPP session over X.25, identify the cloning interface, and configure the interface to initiate AO/DI calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# x25 map` *ppp x121-address* `interface` *cloning-interface*	Enables the interface to initiate a PPP session over the X.25 protocol and remote end mapping.

Enabling the MLP Bundle to Add Multiple Links

Once MLP is enabled and the primary traffic load is reached (based on the **dialer load-threshold** value), the MLP bundle will add member links (B channels). The addition of another B channel places the first link member into "receive-only" mode and subsequent links are added, as needed.

To configure the dialer interface or BRI interface used for cloning purposes and to place the first link member into receive only mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink idle-link`	Configures the interface to enter "receive only" mode so that MLP links are added as needed.

Modifying BACP Default Settings

During BACP negotiation between peers, the called party indicates the number to call for BACP. This number may be in either a national or subscriber format. A national format indicates that the phone number returned from the server to the client should contain ten digits. A subscriber number format contains seven digits.

To assign a prefix to the phone number that is to be returned, use the following optional command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp bap number prefix` *prefix-number*	(Optional) specifies a primary telephone number prefix for a peer to call for PPP BACP negotiation.

Note The **ppp bap number prefix** command is not typically required on the server side, as the server usually does not initiate calls to the client. This command would only be used on the server in a scenario where both sides are configured to act as both client and server.

Configuring the AO/DI Server

The AO/DI server will receive calls from the remote end interface running AO/DI client and likewise, and must be configured to initiate a PPP session over X.25, allow interface cloning, and be capable of adding links to the MLP bundle. The interface configured for AO/DI server relies on the **no-outgoing** option for the **x25 map** command to ensure calls are not originated by the interface. Use the commands in the following sections to configure the AO/DI server:

- Enabling the Interface to Receive AO/DI Client Calls (Required)
- Enabling the MLP Bundle to Add Multiple Links (Required)
- Modifying BACP Default Settings (Optional)

See the section "AO/DI Server Configuration Example" at the end of this chapter for an example of how to configure the AO/DI server.

Enabling the Interface to Receive AO/DI Client Calls

Configure the **x25 map** command with the X.121 address of the calling client. This task enables the AO/DI server interface to run a PPP over X.25 session with the configured client. The **no-outgoing** option must be set in order to ensure that calls do not originate from this interface.

To configure an interface for AO/DI server, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# x25 map ppp` *x121-address* `interface` *cloning-interface* `no-outgoing`	Enables the interface to initiate a PPP session over the X.25 protocol and remote end mapping.

Enabling the MLP Bundle to Add Multiple Links

Once MLP is enabled and the primary traffic load is reached (based on the **dialer load-threshold** value), the MLP bundle will add member links (B channels). The addition of another B channel places the first link member into "receive-only" mode and subsequent links are added, as needed.

To configure the dialer interface or BRI interface used for cloning purposes and to place the first link member into receive only mode, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink idle-link`	Configures the interface to enter "receive only" mode so that MLP links are added as needed.

Modifying BACP Default Settings

During BACP negotiation between peers, the called party indicates the number to call for BACP. This number may be in either a national or subscriber format. A national format indicates that the phone number returned from the server to the client should contain 10 digits. A subscriber number format contains 7 digits.

To assign a prefix to the phone number that is to be returned, use the following, optional command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp bap number {format national | subscriber}`	(Optional) Specifies that the primary telephone number for a peer to call is in either a national or subscriber number format.

Note The **ppp bap number prefix** command is not typically required on the server side, because the server usually does not initiate calls to the client. This command would only be used on the server in a scenario where both sides are configured to act as both client and server.

Configuration Examples for AO/DI

This section provides the following configuration examples:

- AO/DI Client Configuration Example
- AO/DI Server Configuration Example

AO/DI Client Configuration Example

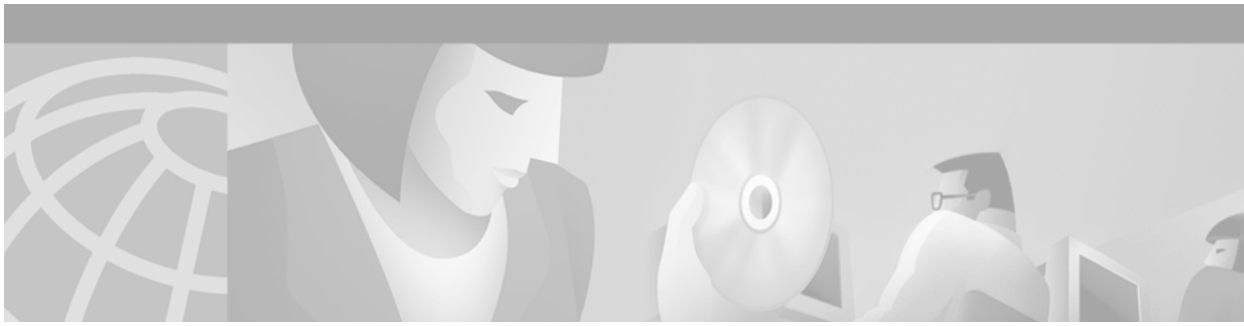
The following example shows BRI interface 0 configured with the PPP, multilink, and X.25 commands necessary for the AO/DI client:

```
hostname Router_client
!
ip address-pool local
isdn switch-type basic-5ess
x25 routing
!
interface Ethernet0
 ip address 172.21.71.99 255.255.255.0
!
interface BRI0
 isdn switch-type basic-5ess
 ip address 10.1.1.9 255.0.0.0
 encap ppp
 dialer in-band
 dialer load-threshold 1 either
 dialer-group 1
 no fair-queue
 ppp authentication chap
 ppp multilink bap
 ppp bap callback accept
 ppp bap call request
 ppp bap number prefix 91
 ppp multilink idle-link
 isdn x25 static-tei 23
 isdn x25 dchannel
 dialer rotary-group 1
!
interface BRI0:0
no ip address
 x25 address 12135551234
 x25 aodi
 x25 htc 4
 x25 win 3
 x25 wout 3
 x25 map ppp 12135556789 interface bri0
!
dialer-list 1 protocol ip permit
```

AO/DI Server Configuration Example

The following example shows the configuration for the AO/DI server, which is configured to only receive calls from the AO/DI client. The configuration uses the **x25 map ppp** command with the **no-outgoing** option, and the **ppp bap number format** command, which implements the **national** format.

```
hostname Router_server
!
ip address-pool local
isdn switch-type basic-5ess
x25 routing
!
interface Ethernet0
 ip address 172.21.71.100 255.255.255.0
!
interface BRI0
 isdn switch-type basic-5ess
 ip address 10.1.1.10 255.0.0.0
 encap ppp
 dialer in-band
 no fair-queue
 dialer load-threshold 1 either
 dialer-group 1
 ppp authentication pap
 ppp multilink bap
 ppp multilink idle-link
 ppp bap number default 2135550904
 ppp bap number format national
 ppp bap call accept
 ppp bap timeout pending 20
 isdn x25 static-tei 23
 isdn x25 dchannel
 dialer rotary-group 1
!
interface BRI0:0
no ip address
 x25 address 12135556789
 x25 htc 4
 x25 win 3
 x25 wout 3
 x25 map ppp 12135551234 interface bri0 no-outgoing
!
dialer-list 1 protocol ip permit
```



Configuring ISDN on Cisco 800 Series Routers

This chapter describes the Common Application Programming Interface (CAPI) and Remote Common Application Programming Interface (RCAPI) feature for the Cisco 800 series routers. This information is included in the following main sections:

- CAPI and RCAPI Overview
- How to Configure RCAPI
- Configuration Examples for RCAPI

The CAPI is an application programming interface standard used to access ISDN equipment connected to ISDN BRIs and ISDN PRIs. RCAPI is the CAPI feature configured remotely from a PC client.

Before you can enable the RCAPI feature on the Cisco 800 series router, the following requirements must be met:

- Cisco 800 series software with RCAPI support is installed on the router.
- CAPI commands are properly configured on the router.
- Both the CAPI local device console and RCAPI client devices on the LAN are correctly installed and configured with RVS-COM client driver software.

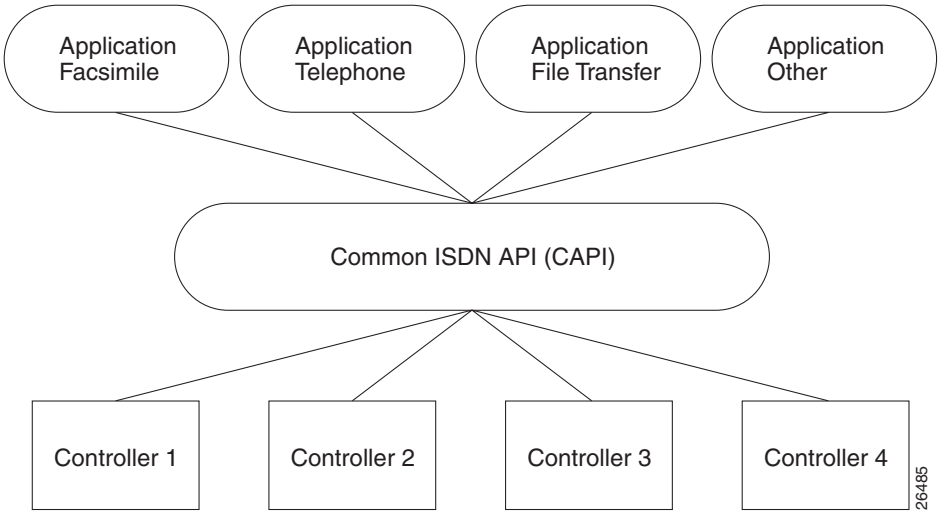
To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release. For more information see the “Identifying Supported Platforms” section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

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

CAPI and RCAPI Overview

Figure 38 shows how CAPI connects applications, drivers, and controllers.

Figure 38 CAPI Connections



Framing Protocols

The framing protocols supported by CAPI include High-Level Data Link Control (HDLC), HDLC inverted, bit transparent (speech), and V.110 synchronous/asynchronous.

Data Link and Network Layer Protocols

CAPI integrates the following data link and network layer protocols:

- Link Access Procedure on the D-channel (LAPD) in accordance with Q.921 for X.25 D-channel implementation
- PPP
- ISO 8208 (X.25 DTE-DTE)
- X.25 DCE, T.90NL, and T.30 (fax group 3)

CAPI Features

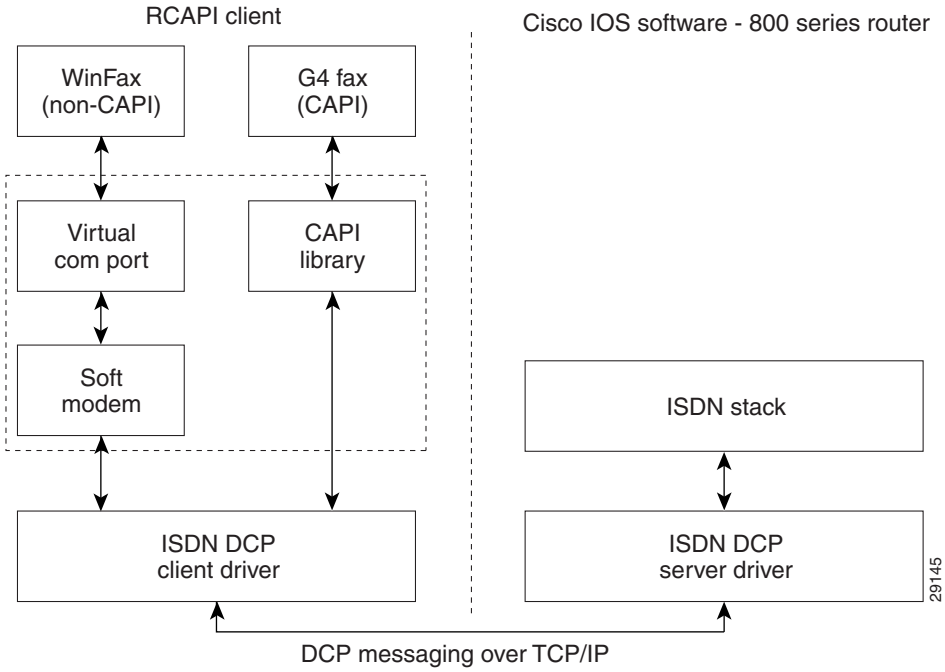
CAPI supports the following features:

- Basic call features, such as call setup and tear-down
- Multiple B channels for data and voice connections
- Multiple logical data link connections within a physical connection
- Selection of different services and protocols during connection setup and on answering incoming calls

- Transparent interface for protocols above Layer 3
- One or more BRIs as well as PRI on one or more Integrated Services Digital Network (ISDN) adapters
- Multiple applications
- Operating-systems-independent messages
- Operating-system-dependent exchange mechanism for optimum operating system integration
- Asynchronous event-driven mechanism, resulting in high throughput
- Well-defined mechanism for manufacturer-specific extensions
- Multiple supplementary services

Figure 39 shows the components of the RCAPI implementation.

Figure 39 Components of RCAPI



CAPI provides a standardized interface through which application programs can use ISDN drivers and controllers. One application can use one or more controllers. Several applications can share one or more controllers.

CAPI supplies a selection mechanism that supports applications that use protocols at different levels and standardized network access. An abstraction from different protocol variables is performed to provide this support. All connection-related data, such as connection state and display messages, is available to the applications at any time.

Supported B-Channel Protocols

The router provides two 64-kbps B channels to RCAPI clients. Each B channel can be configured separately to work in either HDLC mode or bit transparent mode. For CAPI support, layers B2 through B7 protocols are transparent to the applications using these B channels.

The ISDN Core Engine of RVS-COM supports the following B-channel protocols:

- CAPI layer B1
  - 64-kbps with HDLC framing
  - 64-kbps bit transparent operation with byte framing from the network
  - T.30 modem for fax group 3
  - Modem with full negotiation
- CAPI layer B2
  - V.120
  - Transparent
  - T.30 modem for fax group 3
  - Modem with full negotiation
- CAPI layer B3
  - Transparent
  - T.90NL with compatibility to T.70NL according to T.90 Appendix II
  - ISO 8208 (X.25 DTE-DTE) modulo 8 and windows size 2, no multiple logical connections
  - T.30 for fax group 3
  - Modem with full negotiation
- T.30 for fax group 3 (SFF file format [default], sending and receiving up to 14400 bit/s with ECM option, modulations V.17, V.21, V.27ter, V.29)
- Analog modem (sending and receiving up to 14,400 bit/s, modulations V.21, V.22, V.22bis, V.23, V.32, V.32bis)

Supported Switch Types

CAPI and RCAPI support is available only for the ISDN switch type Net3.

CAPI and RVS-COM

The router supports the ISDN Device Control Protocol (ISDN-DCP) from RVS-COM. ISDN-DCP allows a workstation on the LAN or router to use legacy dial computer telephony integration (CTI) applications. These applications include placing and receiving telephone calls and transmitting and receiving faxes.

Using ISDN-DCP, the router acts as a DCP server. By default, the router listens for DCP messages on TCP port number 2578 (the Internet-assigned number for RVS-COM DCP) on its LAN port.

When the router receives a DCP message from a DCP client (connected to the LAN port of the router), the router processes the message and acts on it; it can send confirmations to the DCP clients and ISDN packets through the BRI port of the router.

When the router receives packets destined for one of the DCP clients on its BRI port, the router formats the packet as a DCP message and sends it to the corresponding client. The router supports all the DCP messages specified in the ISDN-DCP specifications defined by RVS-COM.

Supported Applications

ISDN-DCP supports CAPI and non-CAPI applications. Applications are supported that use one or two B channels for data transfer, different HDLC-based protocols, Euro File transfer, or G4 fax; also supported are applications that send bit-transparent data such as A/Mu law audio, G3 fax, analog modem, or analog telephones.

Helpful Website

The following Web link provides answers to frequently asked questions about installing and using RCAPI: http://www.cisco.com/warp/partner/synchronicd/cc/pd/rt/800/prodlit/rcapi_qa.htm

How to Configure RCAPI

To configure RCAPI, perform the tasks in the following sections:

- Configuring RCAPI on the Cisco 800 Series Router (Required)
- Monitoring and Maintaining RCAPI (Optional)
- Troubleshooting RCAPI (Optional)

Configuring RCAPI on the Cisco 800 Series Router

To configure RCAPI on the Cisco 800 series router, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type basic-net3`	Sets the switch type. In this example, the switch type is set to NET3 ISDN, which covers the Euro-ISDN E-DSS1 signaling system and is ETSI-compliant.
Step 2	`Router(config)# rcapi number` *number*	Enters the RCAPI directory number assigned by the ISDN provider for the device. An example command: `rcapi number 12345.`

Step 3	`Router(config)# rcapi server port` *number*	The **rcapi server** command is mandatory for RCAPI to be enabled on the router. The parameter port is optional and is entered only when you need to specify a port number for RCAPI functions. Otherwise, the default port 2578 is used.
		An example command with default port 2578:
		`rcapi server port`
		An example command with port 2000:
		`rcapi server port 2000`
		Configure the same number on both the router and the client PC.

	Command	Purpose
Step 4	`Router(config)# interface bri0`	Configures the ISDN BRI interface and begins interface configuration mode.
Step 5	`Router(config-if)# isdn switch-type basic-net3`	Sets the switch type for the bri0 interface. In this example, the switch type is set to NET3 ISDN, which covers the Euro-ISDN E-DSS1 signaling system and is ETSI-compliant.
Step 6	`Router(config-if)# isdn incoming-voice modem`	Sets the modem as the default handler for incoming voice calls.

Note If required, at each remote device console change to global configuration mode, using the command **configure terminal**, and repeat Step 2 through Step 7 to configure that device.

Monitoring and Maintaining RCAPI

To monitor and maintain RCAPI, use the following command in privileged EXEC mode:

Command	Purpose
`Router# show rcapi status`	Displays RCAPI status.

Troubleshooting RCAPI

To test the RCAPI operation, use the following command in privileged EXEC mode

Command	Purpose
`Router# debug rcapi events`	Starts a background debug program.

Configuration Examples for RCAPI

The following configuration output example shows two Cisco 800 series routers configured for RCAPI:

Router 1

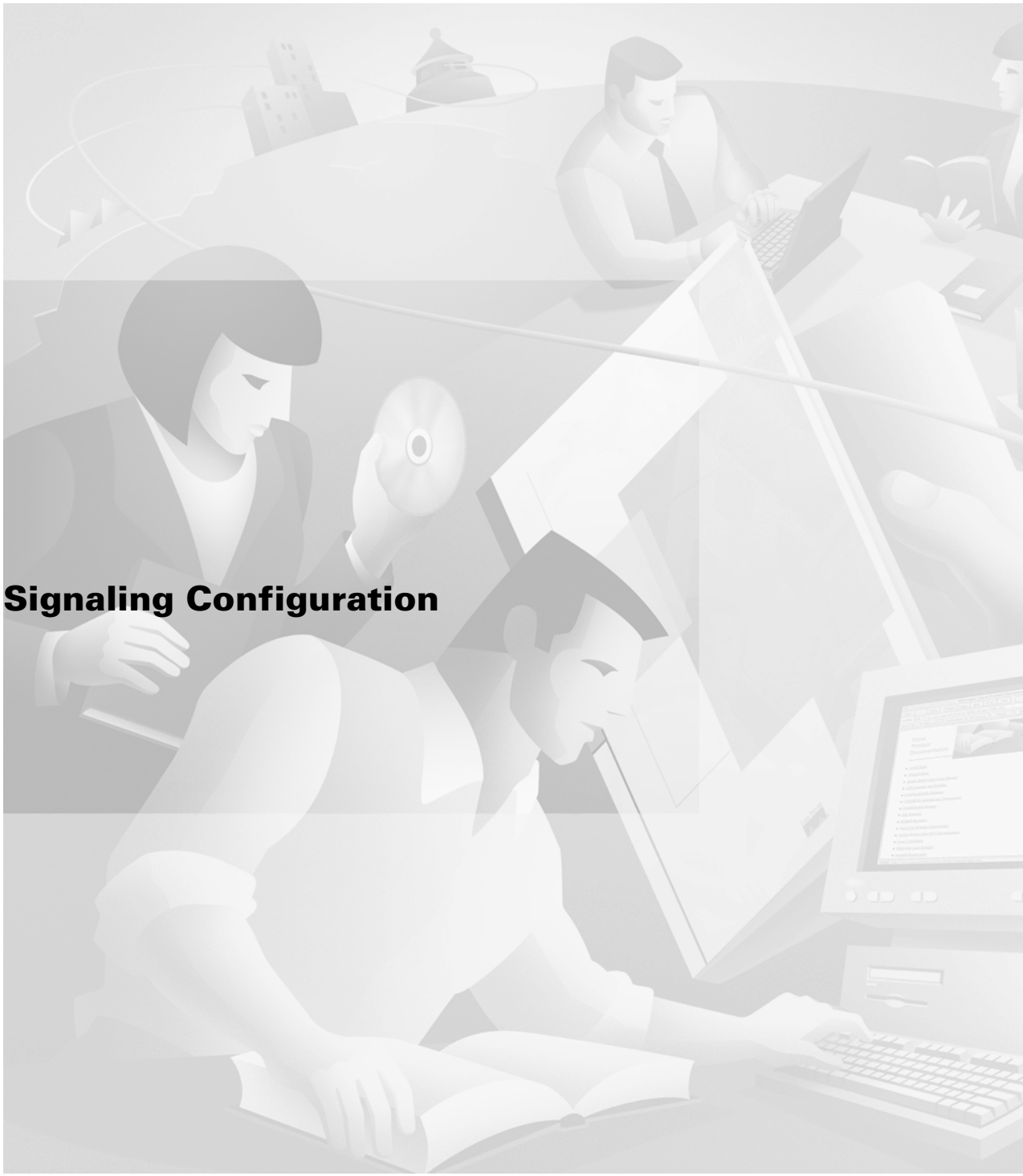
```
Router1# show running-config

Building configuration...

Current configuration:
!
version xx.x
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname local
!
```

```
ip subnet-zero
!
isdn switch-type basic-net3
isdn voice-call-failure 0
!
interface Ethernet0
 ip address 192.168.2.1 255.255.255.0
 no ip directed-broadcast
!
interface BRI0
 no ip address
 no ip directed-broadcast
 isdn switch-type basic-net3
 isdn incoming-voice modem
!
no ip http server
ip classless
!
line con 0
 transport input none
 stopbits 1
line vty 0 4
!
rcapi server port 2578
!
rcapi number 5551000
rcapi number 5553000
!
end

Router1#
```

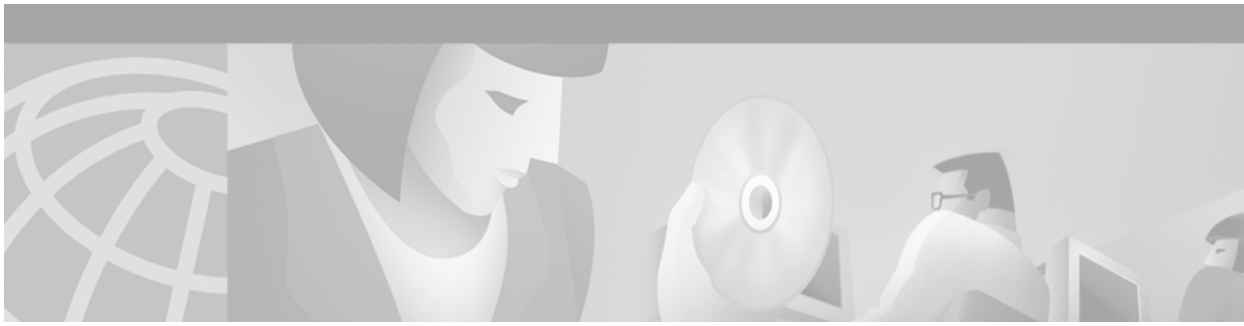
Router 2

```
Router2# show running-config

Building configuration...

Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname local
!
ip subnet-zero
!
isdn switch-type basic-net3
isdn voice-call-failure 0
!
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 no ip directed-broadcast
!
interface BRI0
 no ip address
 no ip directed-broadcast
 isdn switch-type basic-net3
 isdn incoming-voice modem
!
```

```
no ip http server
ip classless
!
line con 0
 transport input none
 stopbits 1
line vty 0
!
rcapi server port 2578
!
rcapi number 5552000
rcapi number 5554000
!
end

Router2#
```



Signaling Configuration



Configuring ISDN PRI

This chapter describes how to configure channelized E1 and channelized T1 for ISDN PRI and for two types of signaling to support analog calls over digital lines. This information is included in the following sections:

- Signaling Overview
- How to Configure ISDN PRI
- Monitoring and Maintaining ISDN PRI Interfaces
- How to Configure Robbed-Bit Signaling for Analog Calls over T1 Lines
- How to Configure CAS
- How to Configure Switched 56K Digital Dial-In over Channelized T1 and Robbed-Bit Signaling
- How to Configure Switched 56K Services
- How to Configure E1 R2 Signaling
- Enabling R1 Modified Signaling in Taiwan
- Configuration Examples for Channelized E1 and Channelized T1

In addition, this chapter describes how to run interface loopback diagnostics on channelized E1 and channelized T1 lines. For more information, see the "How to Configure Switched 56K Digital Dial-In over Channelized T1 and Robbed-Bit Signaling" section later in this chapter, and the *Cisco IOS Interface Configuration Guide*, Release 12.2.

For hardware technical descriptions and for information about installing the controllers and interfaces, refer to the hardware installation and maintenance publication for your particular product.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the channelized E1/T1 commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Signaling Overview

Channelized T1 and channelized E1 can be configured for ISDN PRI, synchronous serial, and asynchronous serial communications.

Channelized T1 and channelized E1 are supported by corresponding controllers. Each T1 or E1 controller has one physical network termination, but it can have many virtual interfaces, depending on the configuration.

In-Band and Out-of-Band Signaling

The terms *in-band* and *out-of-band* indicate whether various signals—which are used to set up, control, and terminate calls—travel in the same channel (or band) with voice calls or data made by the user, or whether those signals travel in a separate channel (or band).

ISDN, which uses the D channel for signaling and the B channels for user data, fits into the out-of-band signaling category.

Robbed-bit signaling, which uses bits from specified frames in the user data channel for signaling, fits into the in-band signaling category.

Channel-associated signaling (CAS), which uses E1 time slot 16 (the D channel) for signaling, fits into the out-of-band signaling category.

Channelized E1 and T1 on Cisco Devices

You can allocate the available channels for channelized E1 or T1 in the following ways:

- All channels can be configured to support ISDN PRI. Channelized T1 ISDN PRI offers 23 B channels and 1 D channel. Channelized E1 ISDN PRI offers 30 B channels and 1 D channel. Channel 24 is the D channel for T1, and channel 16 is the D channel for E1.
- If you are not running ISDN PRI, all channels can be configured to support robbed-bit signaling, which enables a Cisco modem to receive and send analog calls.
- All channels can be configured in a single channel group. For configuration information about this leased line or nondial use, see the "Configuring Serial Interfaces" chapter in the *Cisco IOS Interface Configuration Guide*.
- Mix and match channels supporting ISDN PRI and channel grouping.
- Mix and match channels supporting ISDN PRI, robbed-bit signaling, and channel grouping across the same T1 line. For example, on the same channelized T1 line you can configure the **pri-group timeslots 1-10** command, **channel-group 11 timeslots 11-16** command, and **cas-group 17 timeslots 17-23 type e&m-fgb** command. This is a rare configuration because it requires you to align the correct range of time slots on both ends of the connection.

See the sections "PRI Groups and Channel Groups on the Same Channelized T1 Controller Example," "Robbed-Bit Signaling Examples," and the "ISDN CAS Examples" at the end of this chapter.

How to Configure ISDN PRI

This section describes tasks that are required to get ISDN PRI up and running. This section does not address routing issues, dialer configuration, and dial backup. For information about those topics, see the chapters in the "Dial-on-Demand Routing" part of this manual.

To configure ISDN PRI, perform the tasks in the following sections:

- Requesting PRI Line and Switch Configuration from a Telco Service Provider (Required)
- Configuring Channelized E1 ISDN PRI (As required)
- Configuring Channelized T1 ISDN PRI (As required)
- Configuring the Serial Interface (Required)
- Configuring NSF Call-by-Call Support (Primary-4ESS Only)
- Configuring Multiple ISDN Switch Types (Optional)
- Configuring B Channel Outgoing Call Order (Optional)
- Performing Configuration Self-Tests (Optional)

See the section "Monitoring and Maintaining ISDN PRI Interfaces" later in this chapter for tips on maintaining the ISDN PRI interface. See the end of this chapter for the "ISDN PRI Examples" section.

Note After the ISDN PRI interface and lines are operational, configure the D-channel interface for dial-on-demand routing (DDR). The DDR configuration specifies the packets that can trigger outgoing calls, specifies whether to place or receive calls, and provides the protocol, address, and phone number to use.

Requesting PRI Line and Switch Configuration from a Telco Service Provider

Before configuring ISDN PRI on your Cisco router, you need to order a correctly provisioned ISDN PRI line from your telecommunications service provider.

This process varies dramatically from provider to provider on a national and international basis. However, some general guidelines follow:

- Verify if the outgoing B channel calls are made in ascending or descending order. Cisco IOS default is descending order however, if the switch from the service providers is configured for outgoing calls made in ascending order, the router can be configured to match the switch configuration of the service provider.
- Ask for delivery of calling line identification. Providers sometimes call this CLI or automatic number identification (ANI).
- If the router will be attached to an ISDN bus (to which other ISDN devices might be attached), ask for point-to-multipoint service (subaddressing is required) and a voice-and-data line.

Table 23 provides a sample of the T1 configuration attributes you might request for a PRI switch used in North America.

Table 23 North American PRI Switch Configuration Attributes

Attribute	Value
Line format	Extended Superframe Format (ESF)
Line coding	Binary 8-zero substitution (B8ZS)
Call type	23 incoming channels and 23 outgoing channels
Speed	64 kbps
Call-by-call capability	Enabled
Channels	23 B + D
Trunk selection sequence	Either ascending order (from 1 to 23) or descending order (from 23 to 1)
B + D glare	Yield
Directory numbers	Only 1 directory number assigned by service provider
SPIDs required?	None

Configuring Channelized E1 ISDN PRI

To configure ISDN PRI on a channelized E1 controller, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type switch-type`	Selects a service provider switch type that accommodates PRI. (See Table 24 for a list of supported switch type keywords.)

Step 2	`Router(config)# controller e1 slot/port`
	or
	`Router(config)# controller e1 number`
		Defines the controller location in the Cisco 7200 or Cisco 7500 series router by slot and port number.
		Defines the controller location in the Cisco 4000 series or the Cisco AS5200 universal access server by unit number.[1]

Step 3	`Router(config-controller)# framing crc4`	Defines the framing characteristics as cyclic redundancy check 4 (CRC4).
Step 4	`Router(config-controller)# linecode hdb3`	Defines the line code as high-density bipolar 3 (HDB3).
Step 5	`Router(config-controller)# pri-group [timeslots range]`	Configures ISDN PRI.

1. Controller numbers range from 0 to 2 on the Cisco 4000 series and from 1 to 2 on the Cisco AS5000 series access server.

If you do not specify the time slots, the specified controller is configured for 30 B channels and 1 D channel. The B channel numbers range from 1 to 31; channel 16 is the D channel for E1. Corresponding serial interfaces numbers range from 0 to 30. In commands, the D channel is **interface serial** *controller-number*:**15**. For example, **interface serial 0:15**.

Table 24 lists the keywords for the supported service provider switch types to be used in Step 1 above.

Table 24 ISDN Service Provider PRI Switch Types

Switch Type Keywords	Description/Use
Voice/PBX Systems	
primary-qsig	Supports QSIG signaling per Q.931. Network side functionality is assigned with the **isdn protocol-emulate** command.
Australia and Europe	
primary-net5	NET5 ISDN PRI switch types for Asia, Australia, and New Zealand; ETSI-compliant switches for Euro-ISDN E-DSS1 signaling system.
Japan	
primary-ntt	Japanese NTT ISDN PRI switches.
North America	
primary-4ess	Lucent (AT&T) 4ESS switch type for the United States.
primary-5ess	Lucent (AT&T) 5ESS switch type for the United States.
primary-dms100	Nortel DMS-100 switch type for the United States.
primary-ni	National ISDN switch type.
All Users	
none	No switch defined.

Note For information and examples for configuring ISDN PRI for voice, video, and fax applications, refer to the *Cisco IOS Voice, Video, and Fax Applications Configuration Guide*.

Configuring Channelized T1 ISDN PRI

To configure ISDN PRI on a channelized T1 controller, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type switch-type`	Selects a service provider switch type that accommodates PRI. (Refer to Table 24 for a list of supported PRI switch type keywords.)

Step 2	`Router(config)# controller t1 slot/port`
	or
	`Router(config)# controller t1 number`
		Specifies a T1 controller on a Cisco 7500.
		Specifies a T1 controller on a Cisco 4000.[1]

Step 3	`Router(config-controller)# framing esf`	Defines the framing characteristics as Extended Superframe Format (ESF).

	Command	Purpose
Step 4	`Router(config-controller)# linecode b8zs`	Defines the line code as binary 8 zero substitution (B8ZS).
Step 5	`Router(config-controller)# pri-group [timeslots range]`[2]	Configures ISDN PRI. If you do not specify the time slots, the controller is configured for 23 B channels and 1 D channel.

1. Controller numbers range from 0 to 2 on the Cisco 4000 series and from 1 to 2 on the Cisco AS5000 series.
2. On channelized T1, time slots range from 1 to 24. You can specify a range of time slots (for example, **pri-group timeslots 12-24**) if other time slots are used for non-PRI channel groups.

If you do not specify the time slots, the specified controller is configured for 24 B channels and 1 D channel. The B channel numbers range from 1 to 24; channel 24 is the D channel for T1. Corresponding serial interfaces numbers range from 0 to 23. In commands, the D channel is **interface serial** *controller-number*:**23**. For example, **interface serial 0:23**.

Configuring the Serial Interface

When you configure ISDN PRI on the channelized E1 or channelized T1 controller, in effect you create a serial interface that corresponds to the PRI group time slots. This interface is a logical entity associated with the specific controller. After you create the serial interface by configuring the controller, you must configure the D channel serial interface. The configuration applies to all the PRI B channels (time slots).

To configure the D channel serial interface, perform the tasks in the following sections:

- Specifying an IP Address for the Interface (Required)
- Configuring Encapsulation on ISDN PRI (Required)
- Configuring Network Addressing (Required)
- Configuring ISDN Calling Number Identification (As Required)
- Overriding the Default TEI Value (As Required)
- Configuring a Static TEI (As Required)
- Configuring Incoming ISDN Modem Calls (As Required)
- Filtering Incoming ISDN Calls (As Required)
- Configuring the ISDN Guard Timer (Optional)
- Configuring Inclusion of the Sending Complete Information Element (Optional)
- Configuring ISDN PRI B-Channel Busyout (Optional)

Specifying an IP Address for the Interface

To configure the D channel serial interface created for ISDN PRI, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# interface serial slot/port:23`
	`Router(config)# interface serial number:23`
	
	or
	
	`Router(config)# interface serial slot/port:15`
	`Router(config)# interface serial number:15`
		Specifies D channel on the serial interface for channelized T1 and begins interface configuration mode.
		
		Specifies D channel on the serial interface for channelized E1 and begins interface configuration mode.

Step 2	`Router(config-if)# ip address ip-address`	Specifies an IP address for the interface.

When you configure the D channel, its configuration is applied to all the individual B channels.

Configuring Encapsulation on ISDN PRI

PPP encapsulation is configured for most ISDN communication. However, the router might require a different encapsulation for traffic sent over a Frame Relay or X.25 network, or the router might need to communicate with devices that require a different encapsulation protocol.

Configure encapsulation as described in one of the following sections:

- Configuring PPP Encapsulation
- Configuring Encapsulation for Frame Relay or X.25 Networks
- Configuring Encapsulation for Combinet Compatibility

In addition, the router can be configured for automatic detection of encapsulation type on incoming calls. To configure this feature, complete the tasks in the "Configuring Automatic Detection of Encapsulation Type of Incoming Calls" section.

Note See the sections "Dynamic Multiple Encapsulations" and "Configuring Encapsulation on ISDN BRI" in the chapter "Configuring ISDN BRI" for information about the Cisco Dynamic Multiple Encapsulations feature.

Configuring PPP Encapsulation

Each ISDN B channel is treated as a serial line and supports HDLC and PPP encapsulation. The default serial encapsulation is HDLC. To configure PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Configures PPP encapsulation.

Configuring Encapsulation for Frame Relay or X.25 Networks

If traffic from this ISDN interface crosses a Frame Relay or X.25 network, the appropriate addressing and encapsulation tasks must be completed as required for Frame Relay or X.25 networks.

See the sections "Sending Traffic over Frame Relay, X.25, or LAPB Networks" in the chapter "Configuring Legacy DDR Spokes" for more information about addressing, encapsulation, and other tasks necessary to configure Frame Relay or X.25 networks.

Configuring Encapsulation for Combinet Compatibility

Historically, Combinet devices supported only the Combinet Proprietary Protocol (CPP) for negotiating connections over ISDN B channels. To enable Cisco routers to communicate with those Combinet bridges, the Cisco IOS software supports the CPP encapsulation type.

To enable routers to communicate over ISDN interfaces with Combinet bridges that support only CPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation cpp`	Specifies CPP encapsulation.
Step 2	`Router(config-if)# cpp callback accept`	Enables CPP callback acceptance.
Step 3	`Router(config-if)# cpp authentication`	Enables CPP authentication.

Most Combinet devices support PPP. Cisco routers can communicate over ISDN with these devices by using PPP encapsulation, which supports both routing and fast switching.

Cisco 700 and 800 series routers and bridges (formerly Combinet devices) support only IP, IPX, and bridging. For AppleTalk, Cisco routers automatically perform half-bridging with Combinet devices. For more information about half-bridging, see the section "Configuring PPP Half-Bridging" in the "Configuring Media-Independent PPP and Multilink PPP" chapter in this publication.

Cisco routers can also half-bridge IP and IPX with Combinet devices that support only CPP. To configure this feature, you only need to set up the addressing with the ISDN interface as part of the remote subnet; no additional commands are required.

Configuring Automatic Detection of Encapsulation Type of Incoming Calls

You can enable a serial or ISDN interface to accept calls and dynamically change the encapsulation in effect on the interface when the remote device does not signal the call type. For example, if an ISDN call does not identify the call type in the Lower Layer Compatibility fields and is using an encapsulation that is different from the one configured on the interface, the interface can change its encapsulation type at that time.

This feature enables interoperation with ISDN terminal adapters that use V.120 encapsulation but do not signal V.120 in the call setup message. An ISDN interface that by default answers a call as synchronous serial with PPP encapsulation can change its encapsulation and answer such calls.

Automatic detection is attempted for the first 10 seconds after the link is established or the first 5 packets exchanged over the link, whichever is first.

To enable automatic detection of encapsulation type, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# autodetect encapsulation encapsulation-type`	Enables automatic detection of encapsulation type on the specified interface.

You can specify one or more encapsulations to detect. Cisco IOS software currently supports automatic detection of PPP and V.120 encapsulations.

Configuring Network Addressing

When you configure networking, you specify how to reach the remote recipient. To configure network addressing, use the following commands in interface configuration mode:

	Command	Purpose

Step 1	`Router(config-if)# dialer map protocol next-hop-address name hostname speed 56|64 dial-string[:isdn-subaddress]`
	or
	`Router(config-if)# dialer map protocol next-hop-address name hostname spc [speed 56 | 64] [broadcast] dial-string[:isdn-subaddress]`
		Defines the protocol address of the remote recipient, host name, and dialing string; optionally, provides the ISDN subaddress; sets the dialer speed to 56 or 64 kbps, as needed.
		(Australia) Uses the **spc** keyword that enables ISDN semipermanent connections.

Step 2	`Router(config-if)# dialer-group group-number`	Assigns the interface to a dialer group to control access to the interface.
Step 3	`Router(config-if)# dialer-list dialer-group list access-list-number`	Associates the dialer group number with an access list number.
Step 4	`Router(config-if)# access-list access-list-number {deny | permit} protocol source address source-mask destination destination-mask`	Defines an access list permitting or denying access to specified protocols, sources, or destinations.

Australian networks allow semipermanent connections between customer routers with PRIs and the TS-014 ISDN PRI switches in the exchange. Semipermanent connections are offered at better pricing than leased lines.

Packets that are permitted by the access list specified by the **dialer-list** command are considered interesting and cause the router to place a call to the identified destination protocol address.

Note The access list reference in Step 4 of this task list is an example of the access list commands allowed by different protocols. Some protocols might require a different command form or might require multiple commands. See the relevant chapter in the appropriate network protocol configuration guide (for example, the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*) for more information about setting up access lists for a protocol.

For more information about defining outgoing call numbers, see the sections "Configuring Access Control for Outgoing Calls" in the chapters "Configuring Legacy DDR Spokes" or "Configuring Legacy DDR Hubs" later in this publication.

Configuring ISDN Calling Number Identification

A router might need to supply the ISDN network with a billing number for outgoing calls. Some networks offer better pricing on calls in which the number is presented. When configured, the calling number information is included in the outgoing Setup message.

To configure the interface to identify the billing number, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn calling-number` *calling-number*	Specifies the calling party number.

This command can be used with all ISDN PRI switch types.

Overriding the Default TEI Value

You can configure ISDN terminal endpoint identifier (TEI) negotiation on individual ISDN interfaces. TEI negotiation is useful for switches that may deactivate Layers 1 or 2 when there are no active calls. Typically, this setting is used for ISDN service offerings in Europe and connections to DMS 100 switches that are designed to initiate TEI negotiation.

By default, TEI negotiation occurs when the router is powered up. The TEI negotiation value configured on an interface overrides the default or global TEI value. On PRI interfaces connecting to DMS 100 switches, the router will change the default TEI setting to **isdn tei first-call**. To apply TEI negotiation to a specific PRI interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn tei [first-call | powerup]`	Determines when ISDN TEI negotiation occurs.

Configuring a Static TEI

Depending on the telephone company you subscribe to, you may have a dynamically or statically assigned terminal endpoint identifier (TEI) for your ISDN service. By default, TEIs are dynamic in Cisco routers. To configure the TEI as a static configuration, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn static-tei` *tei-number*	Configures a static ISDN Layer 2 TEI over the D channel.

Configuring Incoming ISDN Modem Calls

All incoming ISDN analog modem calls that come in on an ISDN PRI receive signaling information from the ISDN D channel. The D channel is used for circuit-switched data calls and analog modem calls.

To enable all incoming ISDN voice calls to access the call switch module and integrated modems, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn incoming-voice {modem [56 | 64]}`	Routes incoming ISDN modem calls to the call switch module.

The settings for the **isdn incoming-voice** interface command determine how a call is handled based on bearer capability information, as follows:

- **isdn incoming-voice voice**—Calls bypass the modem and are handled as a voice call.
- **isdn incoming-voice data**—Calls bypass the modem and are handled as digital data.
- **isdn incoming-voice modem**—Calls are passed to the modem and the call negotiates the appropriate connection with the far-end modem.

Refer to the *Cisco IOS Voice, Video, and Fax Configuration Guide* and *Cisco IOS Voice, Video, and Fax Command Reference,* Release 12.2, for more information about using the **isdn incoming-voice** interface configuration command to configure incoming ISDN voice and data calls.

Filtering Incoming ISDN Calls

You may find it necessary to configure your network to reject an incoming call with some specific ISDN bearer capability such as nonspeech or nonaudio data. To filter out unwanted call types, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn reject {{cause cause-code} |{data [56 | 64]} | piafs | v110 | v120 | vod | voice {[3.1khz | 7khz | speech]}}`	Rejects an incoming ISDN BRI or PRI call based on type.

Note When the ISDN interface is configured for incoming voice with the **isdn incoming-voice voice** command (see the previous section "Configuring Incoming ISDN Modem Calls"), and bearer capability indicates the call as unrestricted digital data (i = 0x8890), the call is handled as voice over data (use **vod** keyword).

Verifying the Call Reject Configuration

To verify that calls are being rejected, perform the following steps:

Step 1 Enable the following **debug** commands at the privileged EXEC prompt:

- **debug isdn event**
- **debug isdn event detail**
- **debug isdn q931**
- **debug isdn q931 l3trace**

Step 2 Configure the appropriate **isdn reject** command. The following example configures the network to reject all incoming data calls on ISDN interfaces 4 through 23:

```
Router(config)# interface serial 4:23
Router(config-if)# isdn reject data
Router(config-if)# ^Z
```

Step 3 Build the configuration and then monitor the **debug** command output for the following string, which indicates that the call was rejected:

```
ISDN <TYPE:NUMBER>: Rejecting call id <CALLID> isdn calltype screening failed
```

Step 4 Enter the **show isdn status** EXEC command to display a detailed report of the ISDN configuration, including status of Layers 1 through 3, the call type, and the call identifier.

Step 5 Turn off the debugging messages by entering the **no** form of the **debug** command—**no debug isdn event detail**, for example— or by entering the **undebug** form of the command—**undebug isdn q931**, for example.

Configuring the ISDN Guard Timer

Beginning in Cisco IOS Release 12.2, the ISDN guard timer feature implements a new managed timer for ISDN calls. Because response times for authentication requests can vary, for instance when using DNIS authentication, the guard timer allows you to control the handling of calls.

To configure the ISDN guard timer, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn guard-timer` *msecs*	Enables the guard timer and sets the number of milliseconds for which the access server waits for RADIUS to respond before rejecting or accepting (optional) a call.

For more information about configuring RADIUS, and to see sample ISDN PRI guard timer configurations, refer to the *Cisco IOS Security Configuration Guide*.

Configuring Inclusion of the Sending Complete Information Element

In some geographic locations, such as Hong Kong and Taiwan, ISDN switches require that the Sending Complete information element be included in the outgoing Setup message to indicate that the entire number is included. This information element is generally not required in other locations.

To configure the interface to include the Sending Complete information element in the outgoing call Setup message, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn sending-complete`	Includes the Sending Complete information element in the outgoing call Setup message.

Configuring ISDN PRI B-Channel Busyout

To allow the busyout of individual ISDN PRI B channels, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial` *controller:timeslot*	Enters interface configuration mode for a D-channel serial interface.
Step 2	`Router(config-if)# isdn snmp busyout b-channel`	Allows the busyout of individual PRI B channels via SNMP.

Configuring NSF Call-by-Call Support

Network-Specific Facilities (NSF) are used to request a particular service from the network or to provide an indication of the service being provided. Call-by-call support means that a B channel can be used for any service; its use is not restricted to a certain preconfigured service, such as incoming 800 calls or an outgoing 800 calls. This specific NSF call-by-call service supports outgoing calls configured as voice calls.

This NSF call-by-call support feature is vendor-specific; only routers connected to AT&T Primary-4ESS switches need to configure this feature. This feature is supported on channelized T1.

To enable the router for NSF call-by-call support and, optionally, to place outgoing voice calls, complete the following steps:

Step 1 Configure the controller for ISDN PRI.

Step 2 Configure the D channel interface to place outgoing calls using the **dialer map** command with a **dialing-plan** keyword. You can enter a **dialer map** command for each dialing plan to be supported.

Step 3 Define the dialer map class for that dialing plan.

To define the dialer map class for the dialing plan, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# map-class dialer` *classname*	Specifies the dialer map class, using the **dialing-plan** keyword as the class name, and begins map class configuration mode.
Step 2	`Router(config-map-class)# dialer voice-call`	(Optional) Enables voice calls.
Step 3	`Router(config-map-class)# dialer outgoing` *classname*	Configures the specific dialer map class to make outgoing calls.

Note To set the called party type to international, the dialed number must be prefaced by 011.

Table 25 lists the NSF dialing plans and supported services offered on AT&T Primary-4ESS switches.

Table 25 NSF Supported Services on AT&T Primary-4ESS Switches

NSF Dialing Plan	Data	Voice	International
Software Defined Network (SDN)[1]	Yes	Yes	Global SDN
MEGACOMM	No	Yes	Yes
ACCUNET	Yes	Yes	Yes

1. The dialing plan terminology in this table is defined and used by AT&T.

Configuring Multiple ISDN Switch Types

You can apply an ISDN switch type on a per-interface basis, thus extending the existing global **isdn switch-type** command to the interface level. This allows PRI and BRI to run simultaneously on platforms that support both interface types.

A global ISDN switch type is required and must be configured on the router before you can configure a switch type on an interface.

To configure multiple ISDN switch types for a PRI interface using a channelized E1 or channelized T1 controller, use the following command in global configuration mode:

Command	Purpose
`Router(config)# isdn switch-type switch-type`	Applies a global ISDN switch type.

You must ensure that the ISDN switch type is valid for the ISDN interfaces on the router. Table 24 lists valid ISDN switch types for BRI and PRI interfaces.

Note When you configure an ISDN switch type on the channelized E1 or T1 controller, this switch type is applied to all time slots on that controller. For example, if you configure channelized T1 controller 1:23, which corresponds to serial interface 1, with the ISDN switch type keyword **primary-net5**, then all time slots on serial interface 1 (and T1 controller 1) will use the Primary-Net5 switch type.

The following restrictions apply to the Multiple ISDN Switch Types feature:

- You must configure a global ISDN switch type using the existing **isdn switch-type** global configuration command before you can configure the ISDN switch type on an interface. Because global commands are processed before interface level commands, the command parser will not accept the **isdn switch-type** command on an interface unless a switch type is first added globally. Using the **isdn switch-type** global command allows for backward compatibility.
- If an ISDN switch type is configured globally, but not at the interface level, then the global switch type value is applied to all ISDN interfaces.
- If an ISDN switch type is configured globally and on an interface, the interface level switch type supersedes the global switch type at initial configuration. For example, if the global BRI switch-type keyword **basic-net3** is defined and the interface-level BRI switch-type keyword is **basic-ni**, the National ISDN switch type is the value applied to that BRI interface.

- The ISDN global switch type value is only propagated to the interface level on initial configuration or router reload. If you reconfigure the global ISDN switch type, the new value is not applied to subsequent interfaces. Therefore, if you require a new switch type for a specific interface, you must configure that interface with the desired ISDN switch type.
- If an ISDN global switch type is not compatible with the interface type you are using or you change the global switch type and it is not propagated to the interface level, as a safety mechanism, the router will apply a default value to the interface level, as indicated in Table 26.

Table 26 ISDN PRI and ISDN BRI Global Switch Type Keywords

Global Switch Type	PRI Interface	BRI Interface
primary-4ess	**primary-4ess**	**basic-ni**
primary-5ess	**primary-5ess**	**basic-ni**
primary-dms100	**primary-dms100**	**basic-ni**
primary-net5	**primary-net5**	**basic-net3**
primary-ni	**primary-ni**	**basic-ni**
primary-ntt	**primary-ntt**	**basic-ntt**
primary-qsig	**primary-qsig**	**basic-qsig**
primary-ts014	**primary-ts014**	**basic-ts013**
basic-1tr6	**primary-net5**	**basic-1tr6**
basic-5ess	**primary-ni**	**basic-5ess**
basic-dms100	**primary-ni**	**basic-dms100**
basic-net3	**primary-net5**	**basic-net3**
basic-ni	**primary-ni**	**basic-ni**
basic-ntt	**primary-ntt**	**basic-ntt**
basic-qsig	**primary-qsig**	**basic-qsig**
basic-ts013	**primary-ts014**	**basic-ts013**
basic-vn3	**primary-net5**	**basic-vn3**

If, for example, you reconfigure the router to use global switch type keyword **basic-net3**, the router will apply the **primary-net5** ISDN switch type to PRI interfaces and the **basic-net3** ISDN switch type to any BRI interfaces. You can override the default switch assignment by configuring a different ISDN switch type on the associated interface.

Configuring B Channel Outgoing Call Order

You can configure the router to select the first available B channel in ascending order (channel B1) or descending order (channel B23 for a T1 and channel B30 for an E1). To configure the optional task of selecting B channel order for outgoing calls for PRI interface types, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# isdn bchan-number-order {ascending | descending}`	Enables B channel selection for outgoing calls on a PRI interface (optional).

Before configuring the ISDN PRI on your router, check with your service vendor to determine if the ISDN trunk call selection is configured for ascending or descending order. If there is a mismatch between the router and switch with regard to channel availability, the switch will send back an error message stating the channel is not available. By default, the router will select outgoing calls in descending order.

Performing Configuration Self-Tests

To test the ISDN configuration, use the following EXEC commands as needed. Refer to the *Cisco IOS Debug Command Reference* for information about the **debug** commands.

Command	Purpose
`Router> show controllers t1 slot/port`	Checks Layer 1 (physical layer) of the PRI over T1.
`Router> show controllers e1 slot/port`	Checks Layer 1 (physical layer) of the PRI over E1.
`Router> show isdn status`	Checks the status of PRI channels.
`Router# debug q921`	Checks Layer 2 (data link layer).

`Router# debug isdn events`
or
`Router# debug q931`
or
`Router# debug dialer`
or
`Router> show dialer`
	Checks Layer 3 (network layer).

Monitoring and Maintaining ISDN PRI Interfaces

To monitor and maintain ISDN interfaces, use the following EXEC commands as needed:

Command	Purpose

Cisco 7500 series routers
`Router> show interfaces serial slot/port bchannel channel-number`
or
Cisco 4000 series routers
`Router> show interfaces serial number bchannel channel-number`
	Displays information about the physical attributes of the ISDN PRI over T1 B and D channels.

Cisco 7500 series routers
`Router> show interfaces serial slot/port bchannel channel-number`
or
Cisco 4000 series routers
`Router> show interfaces serial number bchannel channel-number`
	Displays information about the physical attributes of the ISDN PRI over E1 B and D channels.

Cisco 7500 series routers
`Router> show controllers t1 [slot/port]`
or
Cisco 4000 series routers
`Router> show controllers t1 number`
	Displays information about the T1 links supported on the ISDN PRI B and D channels.

Cisco 7500 series routers
`Router> show controllers e1 [slot/port]`
or
Cisco 4000 series routers
`Router> show controllers e1 number`
	Displays information about the E1 links supported on the ISDN PRI B and D channels.

`Router> show isdn {active | history | memory | services | status [dsl | serial number] | timers}`	Displays information about current calls, history, memory, services, status of PRI channels, or Layer 2 or Layer 3 timers. (The **service** keyword is available for PRI only.)
`Router> show dialer [interface type number]`	Obtains general diagnostic information about the specified interface.

How to Configure Robbed-Bit Signaling for Analog Calls over T1 Lines

Some Cisco access servers support robbed-bit signaling for receiving and sending analog calls on T1 lines. Robbed-bit signaling emulates older analog trunk and line in-band signaling methods that are sent in many networks.

In countries that support T1 framing (such as the United States and Canada), many networks send supervisory and signaling information to each other by removing the 8th bit of each time slot of the 6th and 12th frame for superframe (SF) framing. For networks supporting extended superframe (ESF) framing, the 6th, 12th, 18th, and 24th frames are affected. This additional signaling information is added to support channel banks in the network that convert various battery and ground operations on analog lines into signaling bits.

Robbed-bit signaling configured on a Cisco access server enables integrated modems to answer and send analog calls. Robbed bits are forwarded over digital lines. To support analog signaling over T1 lines, robbed-bit signaling must be enabled.

Note The signal type configured on the access server must match the signal type offered by your telco provider. Ask your telco provider which signal type to configure on each T1 controller.

The Cisco access server has two controllers: controller T1 1 and controller T1 0, which must be configured individually.

To configure robbed-bit signaling support for calls made and received, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller t1 0`	Enables the T1 0 controller and begins controller configuration mode.
Step 2	`Router(config-controller)# cablelength long` *dbgain-value dbloss-value*	If the channelized T1 line connects to a smart jack instead of a CSU, sets pulse equalization (use parameter values specified by your telco service provider).
Step 3	`Router(config-controller)# framing esf`	Sets the framing to match that of your telco service provider, which in most cases is *esf*.
Step 4	`Router(config-controller)# linecode b8zs`	Sets the line-code type to match that of your telco service provider, which in most cases is *b8zs*.
Step 5	`Router(config-controller)# clock source line primary`	Configures one T1 line to serve as the primary or most stable clock source line.
Step 6	`Router(config-controller)# cas-group` *channel-number* `timeslots` *range* `type` *signal*	Configures channels to accept voice calls. This step creates interfaces that you can configure.
Step 7	`Router(config-controller)# fdl {att | ansi}`	Sets the facilities data-link exchange standard for the CSU, as specified by your telco service provider.

If you want to configure robbed-bit signaling on the other T1 controller, repeat Steps 1 through 7, making sure in Step 5 to select T1 controller line 1 as the secondary clock source.

If you want to configure ISDN on the other controller, see the section "How to Configure ISDN PRI" in this chapter. If you want to configure channel groupings on the other controller, see the chapter "Configuring Synchronous Serial Ports" in this publication; specify the channel groupings when you specify the interface.

See the section "Robbed-Bit Signaling Examples" at the end of this chapter for configuration examples.

How to Configure CAS

The following sections describe how to configure channel-associated signaling in Cisco networking devices for both channelized E1 and T1 lines:

- CAS on Channelized E1
- CAS on T1 Voice Channels

CAS on Channelized E1

Cisco access servers and access routers support CAS for channelized E1 lines, which are commonly deployed in networks in Latin America, Asia, and Europe. CAS is configured to support channel banks in the network that convert various battery and ground operations on analog lines into signaling bits, which are forwarded over digital lines.

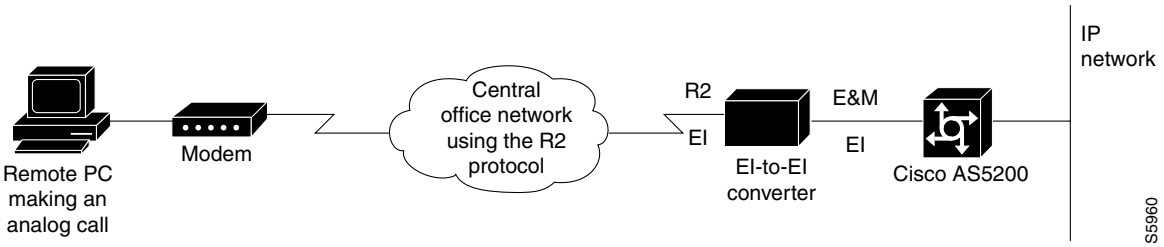
CAS is call signaling that is configured on an E1 controller and enables the access server to send or receive analog calls. The signaling uses the16th channel (time slot); thus, CAS fits in the out-of-band signaling category.

Once CAS is configured on a single E1 controller, remote users can simultaneously dial in to the Cisco device through networks running the R2 protocol (see specifications for your particular network device for the number of dialins supported).

The R2 protocol is an international signaling standard for analog connections. Because R2 signaling is not supported in the Cisco access servers, an E1-to-E1 converter is required.

Figure 40 illustrates that, because the Cisco access servers have more than one physical E1 port on the dual E1 PRI board, up to 60 simultaneous connections can be made through one dual E1 PRI board.

Figure 40 Remote PC Accessing Network Resources Through the Cisco AS5000 Series Access Server



Note For information on how to configure an Anadigicom E1-to-E1 converter, see to the documentation that came with the converter.

Note The dual E1 PRI card must be installed in the Cisco access server before you can configure CAS. To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information.

Configuring CAS for Analog Calls over E1 Lines

To configure the E1 controllers in the Cisco access servers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller e1 number`	Defines the controller location in the Cisco access server by unit number (choices for the *number* argument are 1 or 2) and begins controller configuration mode.
Step 2	`Router(config-controller)# cas-group channel-number timeslots range type signal`	Configures CAS and the R2 signaling protocol on a specified number of time slots.
Step 3	`Router(config-controller)# framing crc4`	Defines the framing characteristics as CRC4.
Step 4	`Router(config-controller)# linecode hdb3`	Defines the line code as HDB3.
Step 5	`Router(config-controller)# clock source line primary`[1]	Specifies one E1 line to serve as the primary or most stable clock source line.

1. Specify the other E1 line as the secondary clock source using the **clock source line secondary** command.

If you do not specify the time slots, CAS is configured on all 30 B channels and one D channel on the specified controller.

See the section "ISDN CAS Examples" for configuration examples.

Configuring CAS on a Cisco Router Connected to a PBX or PSTN

To define E1 channels for the CAS method by which the router connects to a PBX or PSTN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller e1 slot/port`	Specifies the E1 controller that you want to configure with R2 signaling and begins controller configuration.
Step 2	`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 | fxo-loop-start}`	Configures channel-associated signaling and the signaling protocol on a specified number of time slots.
Step 3	`Router(config-controller)# framing crc4`	Defines the framing characteristics as cyclic redundancy check 4 (CRC4).
Step 4	`Router(config-controller)# linecode hdb3`	Defines the line code as high-density bipolar 3 (HDB3).
Step 5	`Router(config-controller)# clock source line primary`[1]	Specifies one E1 line to serve as the primary or most stable clock source line.

1. Specify the other E1 line as the secondary clock source using the **clock source line secondary** command.

If you do not specify the time slots, channel-associated signaling is configured on all 30 B channels and one D channel on the specified controller.

CAS on T1 Voice Channels

Various types of CAS signaling are available in the T1 world. The most common forms of CAS signaling are loop-start, ground-start, and recEive and transMit (E&M). The biggest disadvantage of CAS signaling is its use of user bandwidth to perform signaling functions. CAS signaling is often referred to as robbed-bit-signaling because user bandwidth is being "robbed" by the network for other purposes. In addition to receiving and placing calls, CAS signaling also processes the receipt of DNIS and ANI information, which is used to support authentication and other functions.

This configuration allows the Cisco access servers to provide the automatic number identification/dialed number identification service (ANI/DNIS) delimiter on incoming T1/CAS trunk lines. The digit collection logic in the call switching module (CSM) for incoming T1 CAS calls in dual tone multifrequency (DTMF) is modified to process the delimiters, the ANI digits, and the DNIS digits.

As part of the configuration, a CAS signaling class with the template to process ANI/DNIS delimiters has to be defined. This creates a signaling class structure which can be referred to by its name.

This feature is only functional in a T1 CAS configured for E&M-feature group b (wink start). E&M signaling is typically used for trunks. It is normally the only way that a central office (CO) switch can provide two-way dialing with direct inward dialing. In all the E&M protocols, off-hook is indicated by A=B=1, and on-hook is indicated by A=B=0. If dial pulse dialing is used, the A and B bits are pulsed to indicate the addressing digits.

For this feature, here is an example of configuring for E&M-feature group b:

```
ds0-group 1 timeslots 1-24 type e&m-fgb dtmf dnis
```

In the original Wink Start 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.

Configuring ANI/DNIS Delimiters for CAS Calls on CT1

To configure the signaling class and ANI/DNIS delimiters, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# signaling-class cas name`	Names the signaling class and begins interface configuration mode.
Step 2	`Router(config-if)# profile incoming template`	Defines the template to process the ANI/DNIS delimiter.
Step 3	`Router(config-if)# exit`	Return to global configuration mode.
Step 4	`Router(config)# controller t1 slot/port/number`	Enables this feature for a T1 controller and begins controller configuration mode.
Step 5	`Router(config-controller)# cas-custom channel`	Specifies a single channel group number.
Step 6	`Router(config-ctrl-cas)# class name`	Enables the ANI/DNIS delimiter feature by specifying the template.

To disable the delimiter, use the command **no class** under the cas-custom configuration.

To remove the signaling class, use the configuration command **no signaling-class cas**. When removing a signaling class, make sure the signaling class is no longer used by any controllers; otherwise, the following warning will be displayed:

```
% Can't delete, signaling class test is being used
```

How to Configure Switched 56K Digital Dial-In over Channelized T1 and Robbed-Bit Signaling

Internet service providers (ISPs) can provide switched 56-kbps access to their customers using a Cisco AS5000 series access server. Switched 56K digital dial-in enables many services for ISPs. When using traditional ISDN PRI, the access server uses the bearer capability to determine the type of service. However when providing switched 56K over a CT1 RBS connection, the digital signal level 0 (DS0s) in the access server can be configured to provide either modem or 56-kbps data service. The dial-in user can access a 56-kbps data connection using either an ISDN BRI connection or a 2- or 4-wire switched 56-kbps connection. The telco to which the access server connects must configure its switches to route 56-kbps data calls and voice (modem) calls to the appropriate DS0.

Likewise, an enterprise can provide switched 56-kbps digital dial-in services to its full time telecommuters or small remote offices using ISDN PRI or a CT1 RBS connection.

Switched 56K digital dial-in offers the following benefits:

- Enables ISDN BRI clients to connect to a Cisco access server over switched 56K and T1 CAS.
- Provides switched 56K dial-in services over T1 CAS to remote clients that do not have access to ISDN BRI, for example, a remote PC making digital calls over a 2- or 4-wire switched 56-kbps connection and a CSU.

The following prerequisites apply to the Switched 56K Digital Dial-In feature:

- The remote device could be an ISDN BRI end point such as a terminal adapter or BRI router. In this scenario, the CSU/DSU is irrelevant. For 2- or 4-wire switched 56K remote clients, the remote endpoint must be compatible with the service of the carrier. Different carriers may implement different versions of switched 56K end points.
- A CSU/DSU must be present at the remote client side of the connection. Otherwise, switched 56K connections are not possible. The Cisco access servers have built-in CSU/DSUs.
- The telco must configure its side of the T1 connection to deliver 56-kbps data calls to the correct range of DS0s. If you do not want to dedicate all the DS0s or time slots on a single T1 to switched 56K services, be sure to negotiate with the telco about which DS0s will support switched 56K and which DS0s will not.
- Cisco IOS Release 11.3(2)T or later must be running on the access server.

The following restrictions apply to Switched 56K digital dial-in:

- A Cisco access server only supports incoming switched 56K calls. Dialing out with switched 56K is not supported at this time.
- Switched 56K over E1 is not supported. Only switched 56K over T1 is supported.

- Analog modem calls are not supported over DS0s that are provisioned for switched 56K. For a configuration example, see the section "Switched 56K and Analog Modem Calls over Separate T1 CAS Lines Example" later in this chapter.
- Certain types of T1 lines, such as loop start and ground start, might not support this service. Contact your telco vendor to determine if this feature is available.

Switched 56K Scenarios

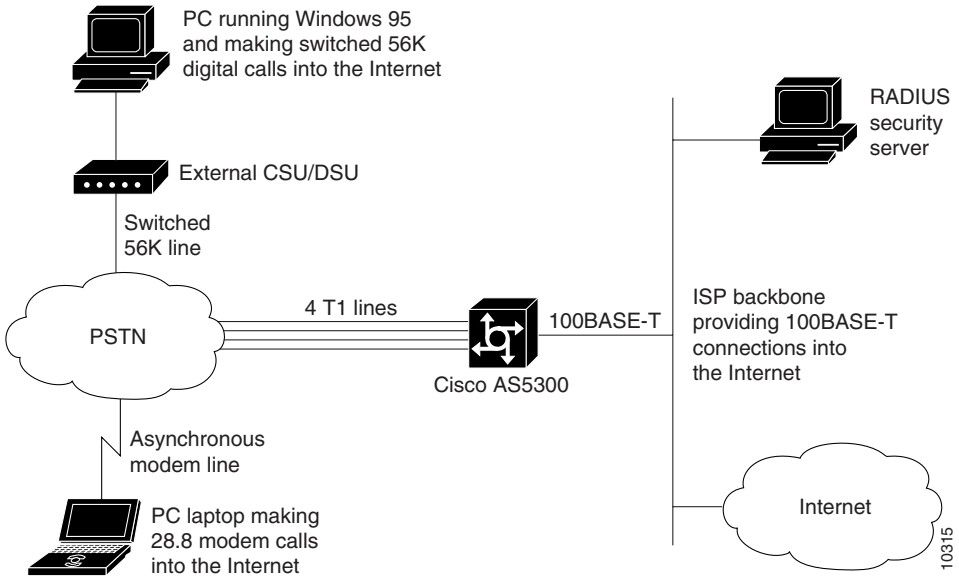
The following scenarios are provided to show multiple applications for supporting switched 56K over T1 CAS:

- Switched 56K and Analog Modem Calls into T1 CAS
- Basic Call Processing Components
- ISDN BRI Calls into T1 CAS

Switched 56K and Analog Modem Calls into T1 CAS

Figure 41 shows a sample network scenario using switched 56K. Two remote PCs are dialing in to the same Cisco access server to get access to the Internet. The desktop PC is making switched 56K digital calls through an external CSU/DSU. The laptop PC is making analog modem calls through a 28.8-kbps modem. The Cisco access server dynamically assigns IP addresses to each node and forwards data packets off to the switched 56K channels and onboard modems respectively.

Figure 41 PCs Making Switched 56K and Analog Modem Calls into a Cisco AS5000 Series Access Server



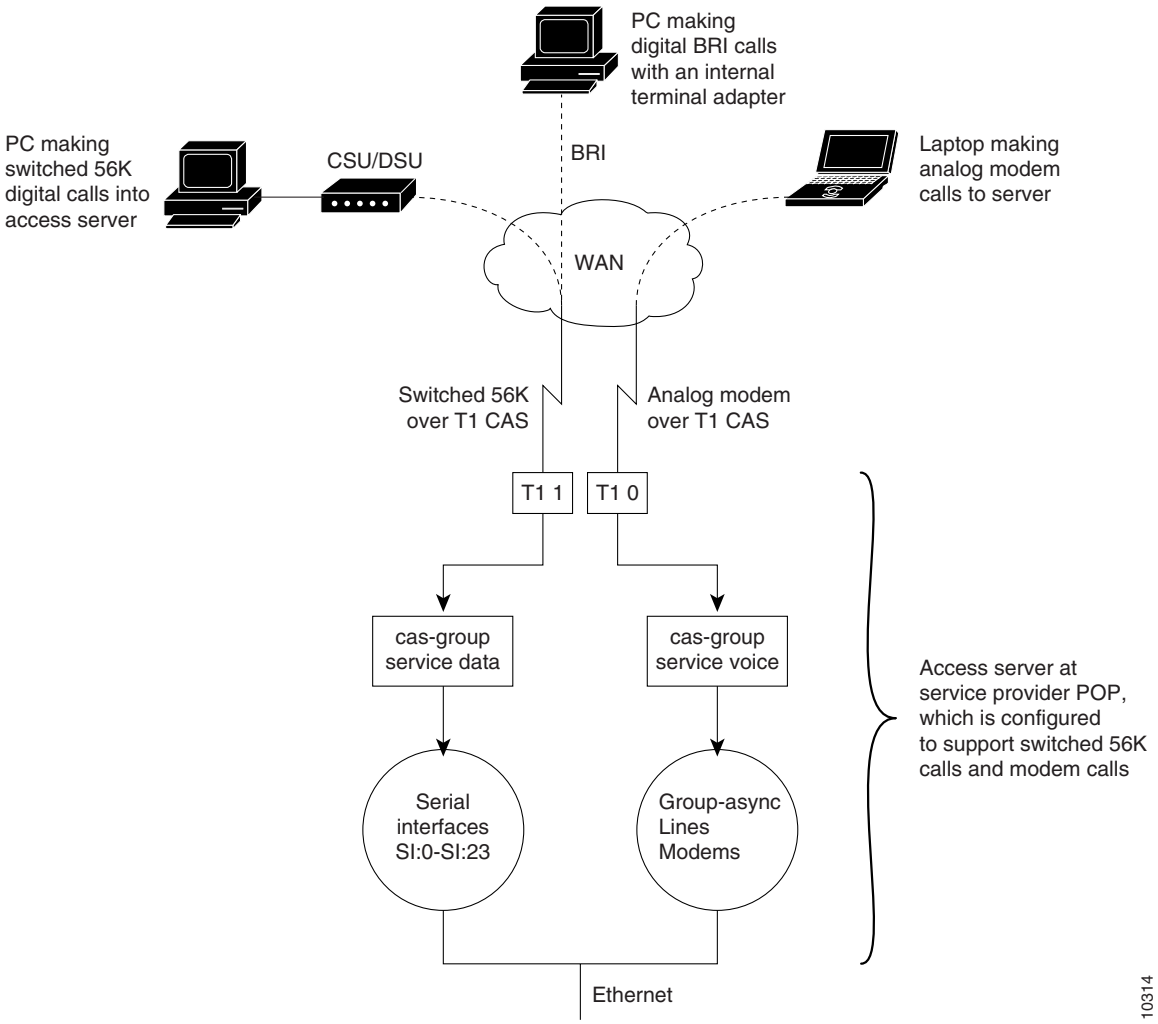
For the startup running configuration on the Cisco access server shown in Figure 41, see the section "Comprehensive Switched 56K Startup Configuration Example" later in this chapter.

Basic Call Processing Components

Figure 42 shows the basic components that process switched 56K calls and analog modem calls on board a Cisco access server. Switched 56K and modem calls are signaling using robbed-bit signaling. Digital switched 56K calls utilize logical serial interfaces just like in ISDN PRI. Modem calls utilize asynchronous interfaces, lines, and modems.

Note The BRI terminal must originate its calls with a bearer capability of 56 kbps.

Figure 42 Processing Components for Switched 56K Calls Versus Analog Modem Calls



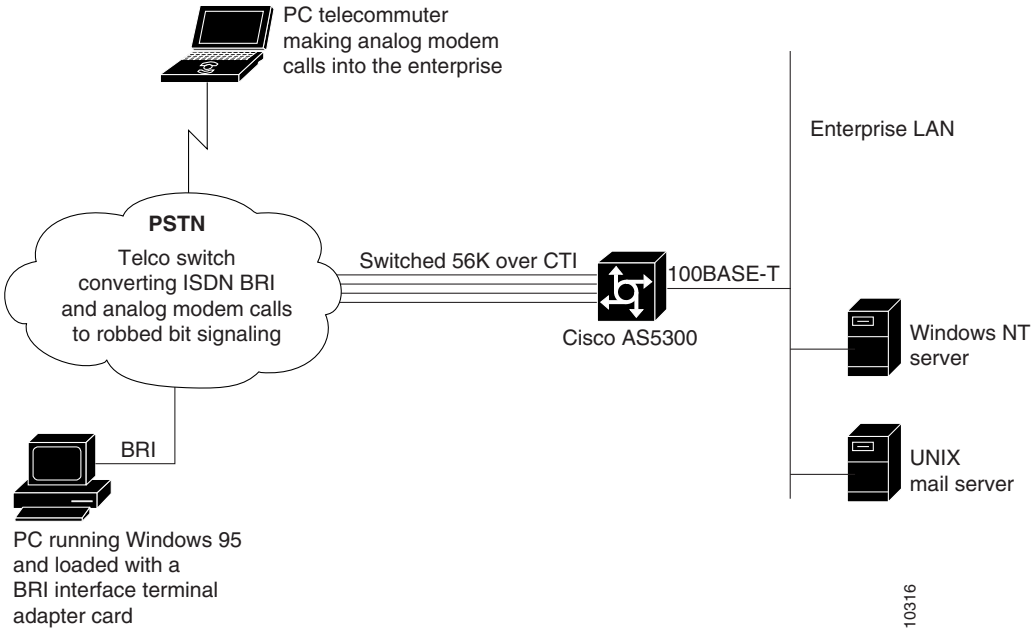
Note The Cisco IOS software does enable you to configure one T1 controller to support both switched 56K digital calls and analog modem calls. In this scenario, Figure 42 would show all calls coming into the access server through one T1 line and controller. However, you must negotiate with the telco which DS0s will support switched 56K services and which DS0s will not. On the access server, analog modem calls are not supported over DS0s that are provisioned for switched 56K. For an example software configuration, see the section "Mixture of Switched 56K and Modem Calls over CT1 CAS Example" at the end of this chapter.

ISDN BRI Calls into T1 CAS

Figure 43 shows how switched 56K functionality can be used to forward ISDN BRI network traffic to a Cisco access server that is configured for switched 56K robbed-bit signaling over CT1.

Note The BRI terminal must originate its calls with a bearer capability of 56 kbps.

Figure 43 Remote PC Making BRI Digital Calls via Switched 56K to a Cisco AS5000 Series Access Server



For a configuration example on the Cisco access server, see the section "Comprehensive Switched 56K Startup Configuration Example" at the end of this chapter.

How to Configure Switched 56K Services

This section describes how to configure switched 56K services on a Cisco access server. After the **cas-group** command is enabled for switched 56K services, a logical serial interface is automatically created for each 56K channel, which must also be configured.

To configure an access server to support switched 56K digital calls, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controllers t1 number`	Specifies a T1 controller and begins controller configuration mode.
Step 2	`Router(config-controller)# framing {sf | esf}`	Sets the framing.
Step 3	`Router(config-controller)# linecode {ami | b8zs}`	Defines the line code.

	Command	Purpose
Step 4	`Router(config-controller)# clock source {line {primary | secondary} | internal}`	Specifies the clocking.
Step 5	`Router(config-controller)# cas-group channel timeslots range type signal`	Configures robbed-bit signaling for a range of time slots. A logical serial interface is automatically created for each switched 56K channel.
Step 6	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 7	`Router(config)# interface serial number:number`	Specifies logical serial interface, which was dynamically created when the **cas-group** command was issued, and configures the core protocol characteristics for the serial interface.

For configuration examples, see the section "Switched 56K Configuration Examples" later in this chapter.

How to Configure E1 R2 Signaling

R2 signaling is an international signaling standard that is common to channelized E1 networks. However, there is no single signaling standard for R2. The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) Q.400-Q.490 recommendation defines R2, but a number of countries and geographic regions implement R2 in entirely different ways. Cisco addresses this challenge by supporting many localized implementations of R2 signaling in its Cisco IOS software.

The following sections offer pertinent information about the E1 R2 signaling feature:

E1 R2 Signaling Overview

R2 signaling is channelized E1 signaling used in Europe, Asia, and South America. It is equivalent to channelized T1 signaling in North America. There are two types of R2 signaling: line signaling and interregister signaling. R2 line signaling includes R2 digital, R2 analog, and R2 pulse. R2 interregister signaling includes R2 compelled, R2 noncompelled, and R2 semicompelled. These signaling types are configured using the **cas-group** command for Cisco access servers, and the **ds0-group** command for Cisco routers.

Many countries and regions have their own E1 R2 variant specifications, which supplement the ITU-T Q.400-Q.490 recommendation for R2 signaling. Unique E1 R2 signaling parameters for specific countries and regions are set by entering the **cas-custom** *channel* command followed by the **country** *name* command.

The Cisco E1 R2 signaling default is ITU, which supports the following countries: Denmark, Finland, Germany, Russia (ITU variant), Hong Kong (ITU variant), and South Africa (ITU variant). The expression "ITU variant" means that there are multiple R2 signaling types in the specified country, but Cisco supports the ITU variant.

Cisco also supports specific local variants of E1 R2 signaling in the following regions, countries, and corporations:

- Argentina
- Australia
- Bolivia[1]
- Brazil
- Bulgaria[1]
- China
- Colombia
- Costa Rica
- East Europe[2]
- Ecuador ITU
- Ecuador LME
- Greece
- Guatemala
- Hong Kong (uses the China variant)
- Indonesia
- Israel
- Korea
- Laos[1]
- Malaysia
- Malta[1]
- New Zealand
- Paraguay
- Peru
- Philippines
- Saudi Arabia
- Singapore
- South Africa (Panaftel variant)
- Telmex corporation (Mexico)
- Telnor corporation (Mexico)
- Thailand
- Uruguay
- Venezuela
- Vietnam

1. Cisco 3620 and 3640 series routers only.
2. Includes Croatia, Russia, and Slovak Republic.

Note Only MICA technologies modems support R2 functionality. Microcom modems do not support R2.

The following are benefits of E1 R2 signaling:

- R2 custom localization—R2 signaling is supported for a wide range of countries and geographical regions. Cisco is continually supporting new countries.
- Broader deployment of dial access services—The flexibility of a high-density access server can be deployed in E1 networks.

Cisco's implementation of R2 signaling has DNIS support turned on by default. If you enable the **ani** option, the collection of DNIS information is still performed. Specifying the **ani** option does not disable DNIS collection. DNIS is the number being called. ANI is the number of the caller. For example, if you are configuring router A to call router B, then the DNIS number is assigned to router B, the ANI number is assigned to router A. ANI is similar to Caller ID.

Figure 44 shows a sample network topology for using E1 R2 signaling with a Cisco AS5800. All four controllers on the access server are configured with R2 digital signaling. Additionally, localized R2 country settings are enabled on the access server.

Figure 44 Service Provider Using E1 R2 Signaling and a Cisco AS5800

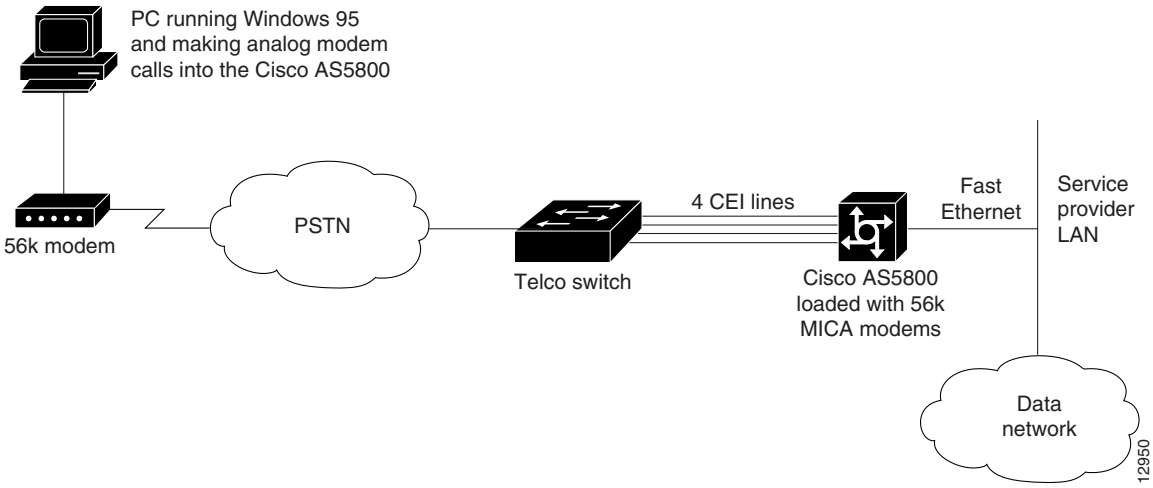
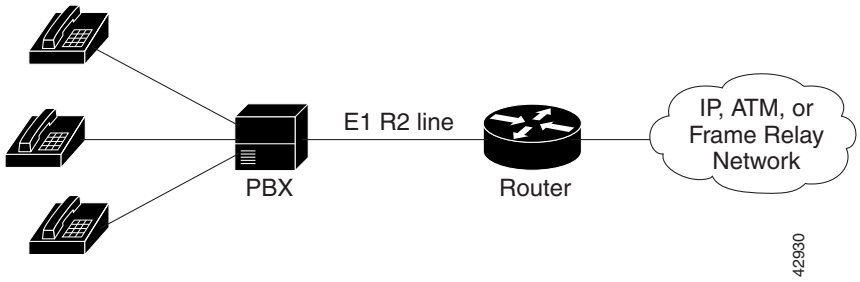


Figure 45 shows a sample network topology for using E1 R2 signaling for voice transfers with a Cisco 2600, 3600, or 7200 series router. All the controllers on the router are configured with R2 digital signaling. Additionally, localized R2 country settings are enabled on the router.

Figure 45 E1 R2 Connections for the Cisco 2600/3600/7200 Series Routers



Configuration examples are supplied in the "Configuration Examples for Channelized E1 and Channelized T1" section at the end of this chapter.

Configuring E1 R2 Signaling

To configure support for E1 R2 signaling on the Cisco access servers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller e1 slot/port`	Specifies the E1 controller that you want to configure with R2 signaling and begins controller configuration mode.
Step 2	`Router(config-controller)# cas-group channel timeslots range type signal` Replace the *signal* argument with any of the following choices under R2 analog, R2 digital, or R2 pulse: **r2-analog** [**dtmf** | **r2-compelled** [**ani**] | **r2-non-compelled** [**ani**] | **r2-semi-compelled** [**ani**]] or **r2-digital** [**dtmf** | **r2-compelled** [**ani**] | **r2-non-compelled** [**ani**] | **r2-semi-compelled** [**ani**]] or **r2-pulse** [**dtmf** | **r2-compelled** [**ani**] | **r2-non-compelled** [**ani**] | **r2-semi-compelled** [**ani**]]	Configures R2 channel associated signaling on the E1 controller. For a complete description of the available R2 options, see the **cas-group** command. The R2 part of this command is defined by the *signal* argument in the **cas-group** command.

For an E1 R2 configuration example, see the section "E1 R2 Signaling Procedure."

Configuring E1 R2 Signaling for Voice

To configure E1 R2 signaling on systems that will be configured for voice, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# controller E1 slot/port`	Specifies the E1 controller that you want to configure with R2 signaling and begins controller configuration mode.
Step 2	`Router(config-controller)# ds0-group channel timeslots range type signal` Replace the signal argument with any of the following choices under R2 analog, R2 digital, or R2 pulse: **r2-analog** [**dtmf** | **r2-compelled** [**ani**] | **r2-non-compelled** [**ani**] | **r2-semi-compelled** [**ani**]] or **r2-digital** [**dtmf** | **r2-compelled** [**ani**] | **r2-non-compelled** [**ani**] | **r2-semi-compelled** [**ani**]] or **r2-pulse** [**dtmf** | **r2-compelled** [**ani**] | **r2-non-compelled** [**ani**] | **r2-semi-compelled** [**ani**]]	Configures R2 channel-associated signaling on the E1 controller. For a complete description of the available R2 options, see the **ds0-group (controller e1)** command reference page.

	Command	Purpose
Step 3	`Router(config-controller)# cas-custom channel`	Enters cas-custom mode. In this mode, you can localize E1 R2 signaling parameters, such as specific R2 country settings for Hong Kong. For the customization to take effect, the *channel* number used in the **cas-custom** command must match the *channel* number specified by the **ds0-group** command.
Step 4	`Router(config-ctrl-cas)# country name use-defaults`	Specifies the local country, region, or corporation specification to use with R2 signaling. Replaces the *name* variable with one of the supported country names. Cisco strongly recommends that you include the **use-defaults** option, which engages the default settings for a specific country. The default setting for all countries is ITU. See the **cas-custom** command reference page for the list of supported countries, regions, and corporation specifications.
Step 5	• `Router(config-ctrl-cas)# ani-digits` • `Router(config-ctrl-cas)# answer-signal` • `Router(config-ctrl-cas)# caller-digits` • `Router(config-ctrl-cas)# category` • `Router(config-ctrl-cas)# default` • `Router(config-ctrl-cas)# dnis-digits` • `Router(config-ctrl-cas)# invert-abcd` • `Router(config-ctrl-cas)# ka` • `Router(config-ctrl-cas)# kd` • `Router(config-ctrl-cas)# metering` • `Router(config-ctrl-cas)# nc-congestion` • `Router(config-ctrl-cas)# unused-abcd` • `Router(config-ctrl-cas)# request-category`	(Optional) Further customizes the R2 signaling parameters. Some switch types require you to fine tune your R2 settings. Do not tamper with these commands unless you fully understand your switch's requirements. For nearly all network scenarios, the **country** *name* **use-defaults** command fully configures your country's local settings. You should not need to perform Step 5. See the **cas-custom** command reference page for more information about each signaling command.

Monitoring E1 R2 Signaling

To monitor E1 R2 signaling, use the following commands in EXEC mode as needed:

Command	Purpose

`Router> show controllers e1`
or
`Router> show controllers e1 number`
	Displays the status for all controllers or a specific controller. Be sure the status indicates the controller is up and there are no alarms or errors (lines 2, 4, 9, and 10, as shown immediately below in the "Monitoring E1 R2 Using the show controllers e1 Command" section).

`Router> show modem csm [slot/port| group number]`	Displays status for a specific modem, as shown below in the "Monitoring E1 R2 Signaling Using the show modem csm Command" section.

Monitoring E1 R2 Using the show controllers e1 Command

```
Router# show controllers e1 0

E1 0 is up.
  Applique type is Channelized E1 - balanced
  No alarms detected.
  Version info of Slot 0:  HW: 2, Firmware: 4, PLD Rev: 2

Manufacture Cookie is not programmed.

  Framing is CRC4, Line Code is HDB3, Clock Source is Line Primary.
  Data in current interval (785 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 13 15 minute intervals):
     0 Line Code Violations, 0 Path Code Violations,
     0 Slip Secs, 12 Fr Loss Secs, 0 Line Err Secs, 0 Degraded Mins,
     0 Errored Secs, 0 Bursty Err Secs, 0 Severely Err Secs, 12 Unavail Secs
```

Monitoring E1 R2 Signaling Using the show modem csm Command

```
Router# show modem csm 1/0

MODEM_INFO: slot 1, port 0, unit 0, tone r2-compelled, modem_mask=0x0000,
modem_port_offset=0
tty_hwidb=0x60E63E4C, modem_tty=0x60C16F04, oobp_info=0x00000000, modem_pool=0x60BC60CC
modem_status(0x0002): VDEV_STATUS_ACTIVE_CALL.
csm_state(0x0205)=CSM_IC5_CONNECTED, csm_event_proc=0x600CFF70, current call thru CAS line
invalid_event_count=0, wdt_timeout_count=0
wdt_timestamp_started is not activated
wait_for_dialing:False, wait_for_bchan:False
pri_chnl=TDM_PRI_STREAM(s0, u3, c7), modem_chnl=TDM_MODEM_STREAM(s1, c0)
dchan_idb_start_index=0, dchan_idb_index=0, call_id=0x0239, bchan_num=6
csm_event=CSM_EVENT_DSX0_CONNECTED, cause=0x0000
ring_no_answer=0, ic_failure=0, ic_complete=3
dial_failure=0, oc_failure=0, oc_complete=0
oc_busy=0, oc_no_dial_tone=0, oc_dial_timeout=0
remote_link_disc=2, stat_busyout=2, stat_modem_reset=0
oobp_failure=0
call_duration_started=00:04:56, call_duration_ended=00:00:00, total_call_duration=00:01:43
The calling party phone number =
The called party phone number  = 9993003
total_free_rbs_timeslot = 0, total_busy_rbs_timeslot = 0, total_dynamic_busy_rbs_timeslot
= 0, total_static_busy_rbs_timeslot = 0, min_free_modem_threshold = 0
```

Verifying E1 R2 Signaling

To verify the E1 R2 signaling configuration, enter the **show controller e1** command to view the status for all controllers, or enter the **show controller e1** *slot/port* command 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 that no alarms (line 6 in the following example) or errors (lines 9, 10, and 11 in the following example) have been reported.

```
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
```

Troubleshooting E1 R2 Signaling

If a connection does not come up, check for the following:

- Loose wires, splices, connectors, shorts, bridge taps, and grounds
- Backward send and receive
- Mismatched framing types (for example, CRC-4 versus no CRC-4)
- Send 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 the following:

- Mismatched line codes (HDB3 versus AMI)
- Receive level
- Frame slips due to poor clocking plan

If problems persist, enable the modem management Call Switching Module (CSM) debug mode, using the **debug modem csm** command, as shown immediately below in the "Debug E1 R1 Signaling Using the debug modem Command" section.

Debug E1 R1 Signaling Using the debug modem Command

```
Router# debug modem csm 1/0

*May 15 04:05:46.675: VDEV_ALLOCATE: slot 2 and port 39 is allocated.

*May 15 04:05:46.675: CSM_RX_CAS_EVENT_FROM_NEAT:(04BF):  EVENT_CALL_DIAL_IN at slot 2 and
port 39

*May 15 04:05:46.675: CSM_PROC_IDLE: CSM_EVENT_DSX0_CALL at slot 2, port 39

*May 15 04:05:46.675: Mica Modem(2/39): Configure(0x0)
*May 15 04:05:46.675: Mica Modem(2/39): Configure(0x3)
*May 15 04:05:46.675: Mica Modem(2/39): Configure(0x6)
*May 15 04:05:46.675: Mica Modem(2/39): Call Setup
*May 15 04:05:46.891: Mica Modem(2/39): State Transition to Call Setup
*May 15 04:05:46.891: Mica Modem(2/39): Went offhook
*May 15 04:05:46.891: CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 2, port 39
```

When the E1 controller comes up, you will see the following messages:

```
%CONTROLLER-3-UPDOWN: Controller E1 0, changed state to up
It also shows these messages for individual timeslots:
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 1 is up
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 2 is up
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 3 is up
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 4 is up
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 5 is up
```

```
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 6 is up
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 7 is up
%DSX0-5-RBSLINEUP: RBS of controller 1 timeslot 8 is up
```

Enabling R1 Modified Signaling in Taiwan

Enabling R1 modified signaling allows a Cisco universal access server to communicate with central office trunks that also use R1 modified signaling. R1 modified signaling is an international signaling standard that is common to channelized T1/E1 networks. Cisco IOS Release 12.1 supports R1 modified signaling customized for Taiwan only. You can configure a channelized T1/E1 interface to support different types of R1 modified signaling, which is used in older analog telephone networks.

This feature allows enterprises and service providers to fully interoperate with the installed Taiwanese telecommunications standards, providing interoperability in addition to the vast array of Cisco IOS troubleshooting and diagnostic capability. This feature will provide customers with a seamless, single-box solution for their Taiwan signaling requirements.

Note This type of signaling is not the same as ITU R1 signaling; it is R1 signaling modified for Taiwan specifically. In the future, R1 modified signaling will be supported by the Cisco AS5800 access server, and will also be available in Turkey.

The following restrictions are for the use of R1 modified signaling:

- Because different line signaling uses different A/B/C/D bit definitions to represent the line state, you must understand the configuration of the T1/E1 trunk before configuring the CAS group. If the wrong type of provision is configured, the access server might interpret the wrong A/B/C/D bit definitions and behave erratically.
- Cisco access servers (Cisco AS5300, and Cisco AS5800) with Microcom modems cannot support this feature.
- You must know the configuration of the T1/E1 trunk before configuring the cas-group. If there is a trunk provisioning mismatch, performance problems may occur.

R1 Modified Signaling Topology

Figure 46 illustrates a service provider using R1 signaling with E1 and a Cisco AS5200 access server. The network topology would be the same for T1 or a Cisco AS5300 access server.

Figure 46 Service Provider Using E1 R1 Signaling with a Cisco AS5200 Access Server

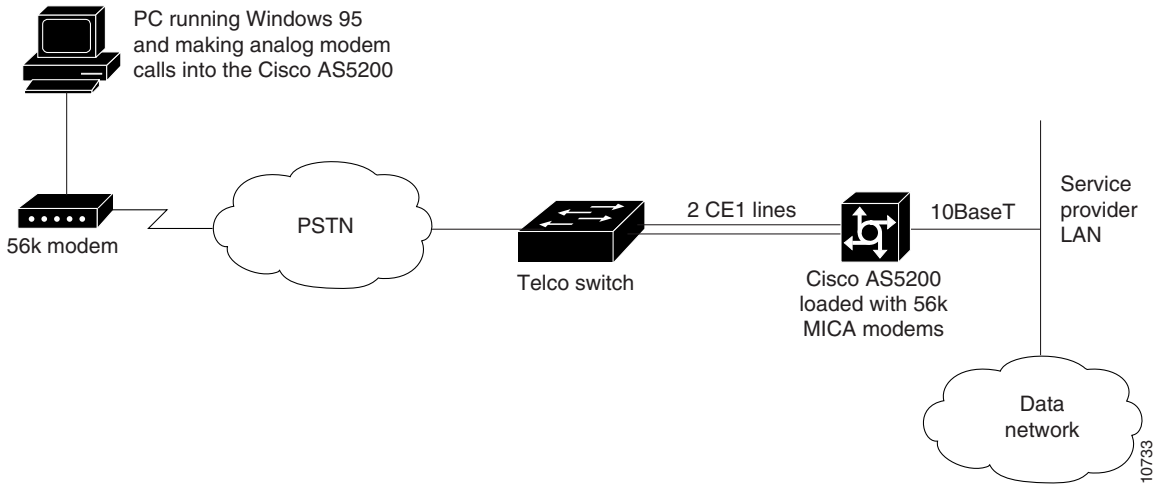
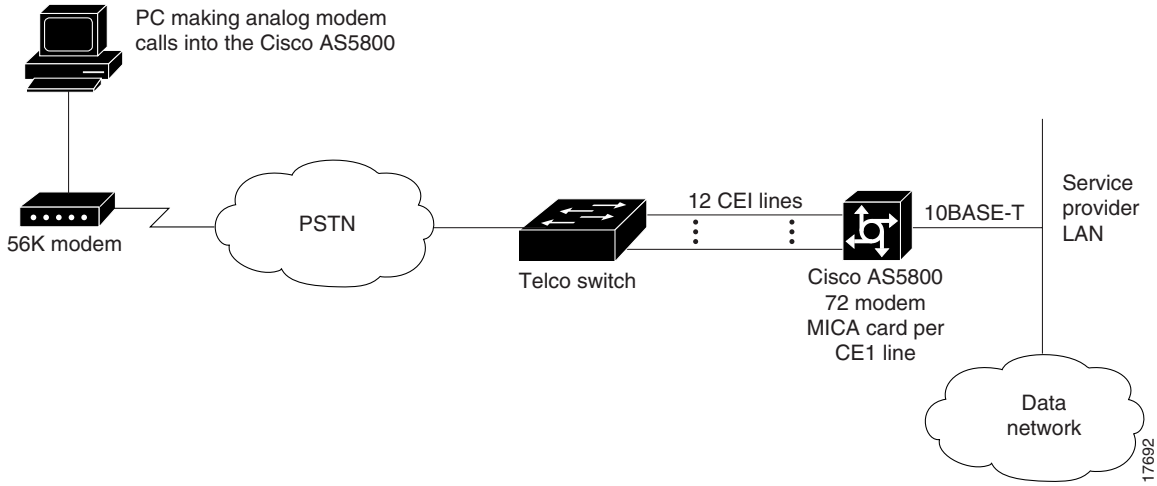


Figure 47 illustrates a service provider using R1 modified signaling with E1 and a Cisco AS5800 access server.

Figure 47 Service Provider Using E1 R1 Modified Signaling with a Cisco AS5800 Access Server



R1 Modified Signaling Configuration Task List

This section describes how to enable R1 modified signaling on your Cisco access server on both a T1 and E1 interface.

Before beginning the tasks in this section, check for the following hardware and software in your system:

- Cisco AS 5200, Cisco AS5300, or Cisco AS5800 access server (without a Microcom modem)
- Cisco IOS Release 12.1 or later software
- MICA feature module
- Portware Version 2.3.1.0 or later

For information on upgrading your Cisco IOS images, modem portware, or modem code, go to the following locations and then select your access server type (Cisco AS5200, Cisco AS5300, or Cisco AS5800) and port information:

- On Cisco.com:
  http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/

  Or, follow this path:
  Cisco Product Documentation/Access Servers and Access Routers/Access Servers
- On the Documentation CD-ROM:
  Cisco Product Documentation/Access Servers and Access Routers/Access Servers

To configure R1 modified signaling, perform the tasks in the following sections, as required:

- Configuring R1 Modified Signaling on a T1 Interface
- Configuring R1 Modified Signaling on an E1 Interface

Note The sample prompts and output are similar for the Cisco AS5200, Cisco AS5300 and Cisco AS5800 access servers.

Configuring R1 Modified Signaling on a T1 Interface

To configure R1 modified signaling on a T1 interface, use the following commands beginning global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5800 access server**
	`Router(config)# vty-async(config)# controller t1 shelf/slot/port`
	`Router(config)# vty-async(config-controller)#`
	
	or
	
	Cisco AS5200 and AS5300 access servers
	`Router(config)# vty-async(config)# controller t1 [0 | 1 | 2 | 3]`
	`Router(config)# vty-async(config-controller)#`
		Specifies the T1 controller that you want to configure and begins controller configuration mode. Refer to the *Cisco AS5800 Universal Access Server Software Installation and Configuration Guide* for port details.
		
		The T1 controller ports are labeled 0 to 3 on the quad T1/PRI cards in the Cisco AS5200 and AS5300 access servers.

Step 2	`Router(config)# vty-async (config-controller)# framing {sf|esf}`	Entering **framing sf** configures framing to T1 with **sf**. Entering **framing esf** configures framing to T1 only.
Step 3	`Router(config)# vty-async (config-controller)# linecode {ami|b8zs}`	Entering **linecode ami** configures line code to AMI[1] encoding. Entering **linecode b8zs** configures line code to **b8zs** encoding.
Step 4	`Router(config)# vty-async (config-controller)# clock source {internal | line [primary | secondary]}`	Entering **clock source internal** configures the clock source to the internal clock. Entering **clock source line primary** configures the clock source to the primary recovered clock. Entering **clock source secondary** configures the clock source to the secondary recovered clock.

	Command	Purpose
Step 5	`Router(config)# vty-async(config-controller)# cas-group 1 timeslots 1-24 type {r1-modified {ani-dnis | dnis} | r1-itu {dnis}}`	Configures the time slots that belong to each E1 circuit for r1-modified or for r1-itu signaling.[2] • The **cas-group** # ranges from 0 to 23 for CT1. • The **timeslot** # ranges from 1 to 24 for CT1. • For the **type**, each CAS group can be configured as one of the Robbed Bit Signaling provisions. • **ani-dnis** indicates R1 will collect ani and dnis information; **dnis** indicates R1 will collect only dnis information.

Step 6	`Router(config)# vty-async(config-if)# ^Z`
	`Router(config)# vty-async#`
	`%SYS-5-CONFIG_I: Configured from console by console`
		Returns to enable mode by simultaneously pressing the **Ctrl** key and the **z** key. (This message returned is expected and does not indicate an error.)

1. AMI = alternate mark inversion.
2. For a more detailed description of the syntax and arguments of this command, refer to the *Cisco IOS Dial Technologies Command Reference*.

Configuring R1 Modified Signaling on an E1 Interface

To configure R1 modified signaling on an E1 interface, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	**Cisco AS5800 access server**
	`Router(config)# controller e1 shelf/slot/port`
	or
	Cisco AS5200 and AS5300 access servers
	`Router(config)# controller e1 [0 | 1 | 2 | 3]`
		Specifies the T1 controller that you want to configure and begins controller configuration mode.
		Refer to the *Cisco AS5800 Universal Access Server Software Installation and Configuration Guide* for port details.
		The T1 controller ports are labeled 0 to 3 on the quad T1/PRI cards in the Cisco AS5200 and AS5300 access servers.

Step 2	`Router (config-controller)# framing {crc4 | no-crc4}`	Entering **framing crc4** configures framing to E1 with CRC.[1] Entering **framing no-crc4** configures framing to E1 only.
Step 3	`Router (config-controller)# linecode {ami | hdb3}`	Entering **linecode ami** configures line code to AMI[2] encoding. Entering **linecode hdb3** configures line code to HDB[3] encoding.
Step 4	`Router (config-controller)# clock source {internal | line [primary | secondary]}`	Entering **clock source internal** configures the clock source to the internal clock. Entering **clock source line primary** configures the clock source to the primary recovered clock. Entering **clock source secondary** configures the clock source to the secondary recovered clock.

	Command	Purpose
Step 5	`Router(config-controller)# cas-group 1 timeslots 1-15, 17-31 type r1-modified {ani-dnis | dnis}`	Configures the time slots that belong to each E1 circuit for R1 modified signaling.[4] • The **cas-group** number ranges from 0 to 30 for CE1. • The **timeslot** number ranges from 1 to 31 for CE1. • For the **type**, each CAS group can be configured as one of the robbed bit signaling provisions. • **ani-dnis** indicates R1 will collect ANI and DNIS information; **dnis** indicates R1 will collect only DNIS information.
Step 6	`Router(config-controller-cas)# cas-custom 1`	(Optional) Enters the channel number to customize.

Step 7	`Router(config-controller-cas)# ^Z`
	`Router#`
	`%SYS-5-CONFIG_I: Configured from console by console`
		Returns to enable mode by simultaneously pressing the **Ctrl** key and the **Z** key.
		
		This message is normal and does not indicate an error.

1. CRC = cyclic redundancy check.
2. AMI = alternate mark inversion.
3. HDB = high-density bipolar 3.
4. For a more detailed description of the syntax and arguments of this command, refer to the *Cisco IOS Dial Technologies Command Reference*.

Troubleshooting Channelized E1 and T1 Channel Groups

Each channelized T1 or channelized E1 channel group is treated as a separate serial interface. To troubleshoot channel groups, first verify configurations and check everything that is normally checked for serial interfaces. You can verify that the time slots and speed are correct for the channel group by checking for CRC errors and aborts on the incoming line.

Note None of the Cisco channelized interfaces will react to any loop codes. To loop a channelized interface requires that the configuration command be entered manually.

Two loopbacks are available for channel groups and are described in the following sections:

- Interface Local Loopback
- Interface Remote Loopback

Interface Local Loopback

Interface local loopback is a bidirectional loopback, which will loopback toward the router and toward the line. The entire set of time slots for the channel group is looped back. The service provider can use a BERT test set to test the link from the central office to your local router, or the remote router can test using pings to its local interface (which will go from the remote site, looped back at your local site, and return to the interface on the remote site).

To place the serial interface (channel group) into local loopback, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# loopback local`	Places the serial interface (channel group) in local loopback.

Interface Remote Loopback

Remote loopback is the ability to put the remote DDS CSU/DSU in loopback. It will work only with channel groups that have a single DS0 (1 time slot), and with equipment that works with a latched CSU loopback as specified in AT&T specification TR-TSY-000476, "OTGR Network Maintenance Access and Testing." To place the serial interface (channel group) in remote loopback, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# loopback remote` *interface*	Places the serial interface (channel group) in remote loopback.

Using the **loopback remote** interface command sends a latched CSU loopback command to the remote CSU/DSU. The router must detect the response code, at which time the remote loopback is verified.

Configuration Examples for Channelized E1 and Channelized T1

ISDN PRI Examples

This section contains the following ISDN PRI examples:

Global ISDN, BRI, and PRI Switch Example

The following example shows BRI interface 0 configured for a NET3 ISDN switch type (**basic-net3** keyword) that will override the National ISDN switch type configured globally. The PRI interface (channelized T1 controller) is configured for ISDN switch type Primary-Net5 and is applied only to the PRI.

```
isdn switch-type basic-ni
!
interface BRI0
 isdn switch-type basic-net3
interface serial0:23
! Apply the primary-net5 switch to this interface only.
 isdn switch-type primary-net5
```

Global ISDN and Multiple BRI and PRI Switch Using TEI Negotiation Example

In the following example, the global ISDN switch type setting is NET3 ISDN (**basic-net3** keyword) and the PRI interface (channelized T1 controller) is configured to use **isdn switch-type primary-net5**. BRI interface 0 is configured for **isdn switch-type basic-ni** and **isdn tei first-call**. TEI first-call negotiation configured on BRI interface 0 overrides the default value (**isdn tei powerup**).

```
isdn switch-type basic-net
!
interface serial0:23
 isdn switch-type primary-net5
 ip address 172.21.24.85 255.255.255.0
!
interface BRI0
 isdn switch-type basic-ni
 isdn tei first-call
```

NSF Call-by-Call Support Example

The following example configures NSF, which is needed for an AT&T 4ESS switch when it is configured for call-by-call support. In call-by-call support, the PRI 4ESS switch expects some AT&T-specific information when placing outgoing ISDN PRI voice calls. The options are **accunet**, **sdn**, and **megacom**.

This example shows both the controller and interface commands required to make the ISDN interface operational and the DDR commands, such as the **dialer map**, **dialer-group**, and **map-class dialer** commands, that are needed to configure the ISDN interface to make outgoing calls.

```
! The following lines configure the channelized T1 controller; all time slots are
! configured for ISDN PRI.
!
controller t1 1/1
 framing esf
 linecode b8zs
 pri-group timeslots 1-23
 isdn switchtype primary-4ess
!
```

```
! The following lines configure the D channel for DDR. This configuration applies
! to all B channels on the ISDN PRI interface.
interface serial 1/1:23
 description Will mark outgoing calls from AT&T type calls.
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name tommyjohn class sdnplan 14193460913
 dialer map ip 10.1.1.3 name angus class megaplan 14182616900
 dialer map ip 10.1.1.4 name angus class accuplan 14193453730

dialer-group 1
 ppp authentication chap

map-class dialer sdnplan
 dialer outgoing sdn

map-class dialer megaplan
 dialer voice-call
 dialer outgoing mega

map-class dialer accuplan
 dialer outgoing accu
```

PRI on a Cisco AS5000 Series Access Server Example

The following example configures ISDN PRI on the appropriate interfaces for IP dial-in on channelized T1:

```
! T1 PRI controller configuration

controller T1 0
 framing esf
 linecode b8zs
 clock source line primary
 pri-group timeslots 1-24
!
controller T1 1
 framing esf
 linecode b8zs
 clock source line secondary
 pri-group timeslots 1-24
!
interface Serial0:23
 isdn incoming-voice modem
 dialer rotary-group 1
!
interface Serial1:23
 isdn incoming-voice modem
 dialer rotary-group 1
!
interface Loopback0
 ip address 172.16.254.254 255.255.255.0
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
!
interface Group-Async1
 ip unnumbered Loopback0
 ip tcp header-compression passive
 encapsulation ppp
 async mode interactive
 peer default ip address pool default
```

```
 dialer-group 1
 ppp authentication chap pap default
 group-range 1 48
!
interface Dialer1
 ip unnumbered Loopback0
 encapsulation ppp
 peer default ip address pool default
 ip local pool default 172.16.254.1 172.16.254.48
 dialer in-band
 dialer-group 1
 dialer idle-timeout 3600
 ppp multilink
 ppp authentication chap pap default
```

The following example configures ISDN PRI on the appropriate interfaces for IP dial-in on channelized E1:

```
! E1 PRI controller configuration

controller E1 0
 framing crc4
 linecode hdb3
 clock source line primary
 pri-group timeslots 1-31
!
controller E1 1
 framing crc4
 linecode hdb3
 clock source line secondary
 pri-group timeslots 1-31

interface serial0:15
 isdn incoming-voice modem
 dialer rotary-group 1
!
interface serial1:15
 isdn incoming-voice modem
 dialer rotary-group 1
!
interface loopback0
 ip address 172.16.254.254 255.255.255.0
!
interface ethernet0
 ip address 172.16.1.1 255.255.255.0
!
! The following block of commands configures DDR for all the ISDN PRI interfaces
! configured above. The dialer-group and dialer rotary-group commands tie the
! interface configuration blocks to the DDR configuration.
!
interface dialer1
 ip unnumbered loopback0
 encapsulation ppp
 peer default ip address pool default
 ip local pool default 172.16.254.1 172.16.254.60
 dialer in-band
 dialer-group 1
 dialer idle-timeout 3600
 ppp multilink
 ppp authentication chap pap default
```

ISDN B-Channel Busyout Example

```
interface Serial0:23
   ip address 172.16.0.0 192.168.0.0
   no ip directed-broadcast
   encapsulation ppp
   no keepalive
   dialer idle-timeout 400
   dialer load-threshold 1 either
   dialer-group 1
   isdn switch-type primary-5ess
   isdn incoming-voice modem
   isdn snmp busyout b-channel
   no fair-queue
   no cdp enable
```

Multiple ISDN Switch Types Example

The following example configures ISDN switch type keyword **primary-4ess** on channelized T1 controller 0 and a switch type keyword **primary-net5** for channelized T1 controller 1.

```
controller t1 0
 framing esf
 linecode b8zs
 isdn switchtype primary-4ess
!
controller t1 1
 framing esf
 linecode b8zs
 isdn switchtype primary-net5
```

The following example shows BRI interface 0 configured for switch type keyword **basic-net3** (NET3 ISDN) that will override the global switch type keyword **basic-ni** (National ISDN). The PRI interface (channelized T1 controller), is configured for ISDN switch type keyword **primary-net5** and is applied only to the PRI interface.

```
isdn switch-type basic-ni
!
interface BRI0
 isdn switch-type basic-net3
interface serial0:23
! Apply the primary-net5 switch to this interface only.
 isdn switch-type primary-net5
```

Outgoing B-Channel Ascending Call Order Example

The following example configures the router to use global ISDN switch-type keyword **primary-ni** and configures the ISDN outgoing call channel selection to be made in ascending order:

```
isdn switch-type primary-ni
!
interface serial0:23
 isdn bchan-number-order ascending
```

Static TEI Configuration Example

The following example shows a static TEI configuration:

```
interface bri 0
 isdn static-tei 1
```

Call Reject Configuration Examples

The following example configures the network to accept incoming ISDN voice calls and reject data calls:

```
interface Serial4:23
  description Connected to V-Sys R2D2
  no ip address
  isdn switch-type primary-5ess
  isdn incoming-voice modem
  isdn reject data
  no cdp enable
 end
```

The following example sets cause code 21 to reject all incoming data calls:

```
interface serial 2/0:23
 isdn reject data
 isdn reject cause 21
```

ISDN Cause Code Override and Guard Timer Example

The following example shows how to configure cause code override and the ISDN guard timer:

```
interface Serial0:23
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer rotary-group 0
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 isdn disconnect-cause 17
 isdn guard-timer 3000 on-expiry accept
 isdn calling-number 8005551234
 no fair-queue
 no cdp enable
```

PRI Groups and Channel Groups on the Same Channelized T1 Controller Example

The following example shows a channelized T1 controller configured for PRI groups and for channel groups. The **pri-group** command and the **channel-group** command cannot have overlapping time slots; note the correct time slot configuration in this example.

```
controller t1 0
channel-group 0 timeslot 1-6
channel-group 1 timeslot 7
channel-group 2 timeslot 8
channel-group 3 timeslot 9-11
pri-group timeslot 12-24
```

The same type of configuration also applies to channelized E1.

Robbed-Bit Signaling Examples

This section provides sample configurations for the T1 controllers on the Cisco access server. You can configure the 24 channels of a channelized T1 to support ISDN PRI, robbed-bit signaling, channel grouping, or a combination of all three. The following samples are provided:

- Allocating All Channels for Robbed-Bit Signaling Example
- Mixing and Matching Channels—Robbed-Bit Signaling and Channel Grouping

Allocating All Channels for Robbed-Bit Signaling Example

The following example configures all 24 channels to support robbed-bit signaling feature group B on a Cisco access server:

```
controller T1 0
 cas-group 1 timeslots 1-24 type e&m-fgb
```

Mixing and Matching Channels—Robbed-Bit Signaling and Channel Grouping

The following example shows you how to configure all 24 channels to support a combination of ISDN PRI, robbed-bit signaling, and channel grouping. The range of time slots that you allocate must match the time slot allocations that your central office chooses to use. This is a rare configuration due to the complexity of aligning the correct range of time slots on both ends of the connection.

The following configuration creates serial interfaces 0 to 9, which correspond to ISDN PRI time slots 1 to 10 (shown as serial 1:0 through serial 1:9). The serial line 1:23 is the D channel, which carries the analog signal bits that dial the phone number of the modem and determine if a modem is busy or available. The D channel is automatically created and assigned to time slot 24.

```
controller T1 0
! ISDN PRI is configured on time slots 1 through 10.
 pri-group timeslots 1-10
! Channelized T1 data is transmitted over time slots 11 through 16.
 channel-group 11 timeslots 11-16
! The channel-associated signal ear and mouth feature group B is configured on
! virtual signal group 17 for time slots 17 to 23, which are used for incoming
! and outgoing analog calls.
 cas-group 17 timeslots 17-23 type e&m-fgb
```

There is no specific interface, such as the serial interface shown in the earlier examples, that corresponds to the time-slot range.

Switched 56K Configuration Examples

The following switched 56K configuration examples are provided:

- Switched 56K T1 Controller Procedure
- Mixture of Switched 56K and Modem Calls over CT1 CAS Example
- Switched 56K and Analog Modem Calls over Separate T1 CAS Lines Example
- Comprehensive Switched 56K Startup Configuration Example

Switched 56K T1 Controller Procedure

The following procedure shows how to configure one T1 controller on a Cisco access server to support switched 56K digital calls. The Cisco access server has four controllers, which are numbered 0 to 3. If you want all four T1 controllers to support switched 56K calls, then repeat this procedure on each controller.

Step 1 Enter global configuration mode using the **configure terminal** command:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Specify a T1 controller with the **controller t1** *number* command. Replace the *number* variable with a controller number from 0 to 3.

```
Router(config)# controller t1 1
```

Step 3 Configure robbed-bit signaling on a range of time slots, then specify switched 56K digital services using the **cas-group** command. In this example, all calls coming into controller T1 1 are expected to be switched 56K data calls, not analog modem calls.

```
Router(config-controller)# cas-group 1 timeslots 1-24 type e&m-fgb service data
```

Note Be sure your signaling type matches the signaling type specified by the central office or telco on the other end. For a list of supported signaling types and how to collect DNIS, refer to the **cas-group** command description for the E1 controller card in the *Cisco IOS Dial Technologies Command Reference*, Release 12.2.

Step 4 Set the framing for your network environment. You can choose ESF (enter **framing esf**) or SF (enter **framing sf**).

```
Router(config-controller)# framing esf
```

Step 5 Set the line-code type for your network environment. You can choose AMI encoding (enter **linecode ami**) or B8ZS encoding (enter **linecode b8zs**).

```
Router(config-controller)# linecode b8zs
```

Mixture of Switched 56K and Modem Calls over CT1 CAS Example

The following example configures one T1 controller to accept incoming switched 56K digital calls and analog modem calls over the same T1 CAS line. Time slots 1 through 10 are provisioned by the telco to support switched 56K digital calls. Time slots 11 through 24 are provisioned to support analog modem calls. Due to the DS0s provisioning, it is impossible for analog modems calls to be sent over the DS0s that map to time slots 1 through 10.

```
controller T1 0
 cas-group 1 timeslots 1-10 type e&m-fgb service data
 cas-group 1 timeslots 11-24 type e&m-fgb service voice
 framing esf
 clock source line primary
 linecode b8zs
 exit
```

Switched 56K and Analog Modem Calls over Separate T1 CAS Lines Example

The following example configures one Cisco access server to accept 50 percent switched 56K digital calls and 50 percent analog modem calls. The controllers T1 0 and T1 1 are configured to support the switched 56K digital calls using the **cas-group 1 timeslots 1-24 type e&m-fgb service digital** command. Controllers T1 2 and T1 3 are configured to support analog modem calls.

```
controller T1 0
 cas-group 1 timeslots 1-24 type e&m-fgb service data
 framing esf
 clock source line primary
 linecode b8zs
 exit
controller T1 1
 cas-group 1 timeslots 1-24 type e&m-fgb service data
 framing esf
 clock source line secondary
 linecode b8zs
 exit
controller T1 2
 cas-group 1 timeslots 1-24 type e&m-fgb service voice
 framing esf
 clock source internal
 linecode b8zs
 exit
controller T1 3
 cas-group 1 timeslots 1-24 type e&m-fgb service voice
 framing esf
 clock source internal
 linecode b8zs
 exit
copy running-config startup-config
```

Comprehensive Switched 56K Startup Configuration Example

The startup configuration in this section runs on the Cisco access server, as shown in Figure 41. This configuration is for an IP dial-in scenario with a mix of switched 56K calls and modem calls. Switched 56K digital calls come into controllers T1 0 and T1 1. Analog modem calls come into controllers T1 2 and T1 3.

In this example, the switched 56K clients are single endpoints in a remote node configuration. If each switched 56K client were instead a router with a LAN behind it without port address translation (PAT) turned on, then a static address, subnet mask, and route must be configured for each remote endpoint. This configuration would best done through RADIUS.

After a T1 time slot is configured with robbed-bit signaling using the **cas-group** command with the **service data** option, a logical serial interface is instantly created for each switched 56K channel. For example, signaling configured on all 24 time slots of controller T1 1 dynamically creates serial interfaces S0:0 through S0:23. You must then configure protocol support on each serial interface. No **interface group** command exists for serial interfaces, unlike asynchronous interfaces via the **interface group-async** command. Each serial interface must be individually configured. In most cases, the serial configurations will be identical. To streamline or shorten this configuration task, you might consider using a dialer interface, as shown in the following example.

Note In the following example, only analog modem calls encounter the group asynchronous and line interfaces. Switched 56K calls encounter the logical serial interfaces and dialer interface.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname 5300
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin radius
aaa authentication ppp default local
aaa authentication ppp dialin if-needed radius
aaa authorization exec local radius
aaa authorization network radius
aaa accounting network start-stop radius
aaa accounting exec start-stop radius
!
enable secret cisco
!
username admin password cisco
async-bootp dns-server 10.1.3.1 10.1.3.2
!
!
! Switched 56K calls come into controllers T1 0 and T1 1. Take note of the keywords
! "service data" in the cas-group command.
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 cas-group 0 timeslots 1-24 type e&m-fgb service data
!
controller T1 1
 framing esf
 clock source line secondary
 linecode b8zs
 cas-group 1 timeslots 1-24 type e&m-fgb service data
!
! Analog modem calls come into controllers T1 2 and T1 3.
!
controller T1 2
 framing esf
 clock source line internal
 linecode b8zs
 cas-group 2 timeslots 1-24 type e&m-fgb
!
controller T1 3
 framing esf
 clock source line internal
 linecode b8zs
 cas-group 3 timeslots 1-24 type e&m-fgb
!
interface loopback0
 ip address 10.1.2.62 255.255.255.192
!
interface Ethernet0
 no ip address
 shutdown
!
```

```
interface FastEthernet0
 ip address 10.1.1.11 255.255.255.0
 ip summary address eigrp 10.10.1.2.0 255.255.255.192
!
! Interface serial0:0 maps to the first switched 56K channel. The dialer pool-member
! command connects this channel to dialer interface 1.
!
interface Serial0:0
 dialer rotary-group 1
!
interface Serial0:1
 dialer rotary-group 1
!
interface Serial0:2
 dialer rotary-group 1
!
interface Serial0:3
 dialer rotary-group 1
!
interface Serial0:4
 dialer rotary-group 1
!
interface Serial0:5
 dialer rotary-group 1
!
interface Serial0:6
 dialer rotary-group 1
!
interface Serial0:7
 dialer rotary-group 1
!
interface Serial0:8
 dialer rotary-group 1
!
interface Serial0:9
 dialer rotary-group 1
!
interface Serial0:10
 dialer rotary-group 1
!
interface Serial0:11
 dialer rotary-group 1
!
interface Serial0:12
 dialer rotary-group 1
!
interface Serial0:13
 dialer rotary-group 1
!
interface Serial0:14
 dialer rotary-group 1
!
interface Serial0:15
 dialer rotary-group 1
!
interface Serial0:16
 dialer rotary-group 1
!
interface Serial0:17
 dialer rotary-group 1
!
interface Serial0:18
 dialer rotary-group 1
!
```

```
interface Serial0:19
 dialer rotary-group 1
!
interface Serial0:20
 dialer rotary-group 1
!
interface Serial0:21
 dialer rotary-group 1
!
interface Serial0:22
 dialer rotary-group 1
!
! Interface serial 0:23 is the last switched 56K channel for controller T1 0.
!
interface Serial0:23
 dialer rotary-group 1
!
! The switched 56K channels for controller T1 1 begin with interface serial 1:0 and end
! with interface serial 1:23.
!
interface Serial1:0
 dialer rotary-group 1
!
interface Serial1:1
 dialer rotary-group 1
!
interface Serial1:2
 dialer rotary-group 1
!
interface Serial1:3
 dialer rotary-group 1
!
interface Serial1:4
 dialer rotary-group 1
!
interface Serial1:5
 dialer rotary-group 1
!
interface Serial1:6
 dialer rotary-group 1
!
interface Serial1:7
 dialer rotary-group 1
!
interface Serial1:8
 dialer rotary-group 1
!
interface Serial1:9
 dialer rotary-group 1
!
interface Serial1:10
 dialer rotary-group 1
!
interface Serial1:11
 dialer rotary-group 1
!
interface Serial1:12
 dialer rotary-group 1
!
interface Serial1:13
 dialer rotary-group 1
!
interface Serial1:14
 dialer rotary-group 1
```

```
!
interface Serial1:15
 dialer rotary-group 1
!
interface Serial1:16
 dialer rotary-group 1
!
interface Serial1:17
 dialer rotary-group 1
!
interface Serial1:18
 dialer rotary-group 1
!
interface Serial1:19
 dialer rotary-group 1
!
interface Serial1:20
 dialer rotary-group 1
!
interface Serial1:21
 dialer rotary-group 1
!
interface Serial1:22
 dialer rotary-group 1
!
interface Serial1:23
 dialer rotary-group 1
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 96
!
interface Dialer1
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.96
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
radius-server host 10.1.1.23 auth-port 1645 acct-port 1646
radius-server host 10.1.1.24 auth-port 1645 acct-port 1646
radius-server key cisco
!
line con 0
 login authentication console
line 1 96
 autoselect ppp
```

```
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

ISDN CAS Examples

This section provides channelized E1 sample configurations for the Cisco access server. You can configure the 30 available channels with CAS, channel grouping, or a combination of the two. The following examples are provided:

- Allocating All Channels for CAS Example
- Mixing and Matching Channels—CAS and Channel Grouping Example

Allocating All Channels for CAS Example

The following interactive example configures channels (also known as time slots) 1 to 30 with ear and mouth channel signaling and feature group B support on a Cisco access server; it also shows that the router displays informative messages about each time slot. signaling messages are sent in the 16th time slot; therefore, that time slot is not brought up.

```
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller e1 0
Router(config-controller)# cas-group 1 timeslots 1-31 type e&m-fgb
Router(config-controller)#
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 1 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 2 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 3 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 4 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 5 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 6 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 7 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 8 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 9 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 10 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 11 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 12 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 13 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 14 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 15 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 17 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 18 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 19 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 20 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 21 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 22 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 23 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 24 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 25 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 26 is up
```

```
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 27 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 28 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 29 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 30 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 31 is up
```

Mixing and Matching Channels—CAS and Channel Grouping Example

The following interactive example shows you how to configure an E1 controller to support a combination of CAS and channel grouping. The range of time slots that you allocate must match the time slot allocations that your central office chooses to use. This configuration is rare because of the complexity of aligning the correct range of time slots on both ends of the connection.

Time slots 1 through 15 are assigned to channel group 1. In turn, these time slots are assigned to serial interface 0 and virtual channel group 1 (shown as serial 0:1).

```
Router(config)# controller e1 0
Router(config-controller)# channel-group 1 timeslots 1-15
Router(config-controller)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:1, changed state to down
%LINK-3-UPDOWN: Interface Serial0:1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:1, changed state to up
```

Time slots 17 to 31 are configured with CAS:

```
Router(config-controller)# cas-group 2 timeslots 17-31 type e&m-fgb
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:1, changed state to down
Router(config-controller)#
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 17 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 18 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 19 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 20 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 21 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 22 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 23 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 24 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 25 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 26 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 27 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 28 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 29 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 30 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 31 is up
Router(config-controller)#
```

E1 R2 Signaling Procedure

The following procedure configures R2 signaling and customizes R2 parameters on controller E1 2 of a Cisco AS5300 access server. In most cases, the same R2 signaling type is configured on each E1 controller.

Step 1 Enter global configuration mode using the **configure terminal** command:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

Step 2 Specify the E1 controller that you want to configure with R2 signaling using the **controller e1** *number* global configuration command. A controller informs the access server how to distribute or provision individual time slots for a connected channelized E1 line. You must configure one E1 controller for each E1 line.

```
Router(config)# controller e1 2
```

Step 3 Configure CAS with the **cas-group** *channel* **timeslots** *range* **type** *signal* command. The signaling type forwarded by the connecting telco switch must match the signaling configured on the Cisco AS5300 access server. The Cisco IOS configuration options are **r2-analog**, **r2-digital**, or **r2-pulse**.

```
Router(config-controller)# cas-group 1 timeslots 1-31 type ?
  e&m-fgb              E & M Type II FGB
  e&m-fgd              E & M Type II FGD
  e&m-immediate-start  E & M Immediate Start
  fxs-ground-start     FXS Ground Start
  fxs-loop-start       FXS Loop Start
  p7                   P7 Switch
  r2-analog            R2 ITU Q411
  r2-digital           R2 ITU Q421
  r2-pulse             R2 ITU Supplement 7
  sas-ground-start     SAS Ground Start
  sas-loop-start       SAS Loop Start
```

The following example specifies R2 ITU Q421 digital line signaling (**r2-digital**). This example also specifies R2 compelled register signaling and provisions the ANI ADDR option.

```
Router(config-controller)# cas-group 1 timeslots 1-31 type r2-digital r2-compelled ani
Router(config-controller)#
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 1 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 2 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 3 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 4 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 5 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 6 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 7 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 8 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 9 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 10 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 11 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 12 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 13 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 14 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 15 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 17 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 18 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 19 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 20 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 21 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 22 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 23 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 24 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 25 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 26 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 27 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 28 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 29 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 30 is up
%DSX0-5-RBSLINEUP: RBS of controller 0 timeslot 31 is up
```

Note The actual R2 CAS is configured on the 16th time slot, which is why the time slot does not come up in the example output. For a description of the supported R2 signaling options, refer to the **cas-group** command for the E1 controller in the *Cisco IOS Dial Technologies Command Reference*.

Step 4 Customize some of the E1 R2 signaling parameters with the **cas-custom** *channel* controller configuration command. This example specifies the default R2 settings for Argentina. For custom options, refer to the **cas-custom** command in the *Cisco IOS Dial Technologies Command Reference*.

```
Router(config-controller)# cas-custom 1
Router(config-ctrl-cas)# ?
CAS custom commands:
  ani-digits        Expected number of ANI digits
  answer-signal     Answer signal to be used
  caller-digits     Digits to be collected before requesting CallerID
  category          Category signal
  country           Country Name
  default           Set a command to its defaults
  dnis-digits       Expected number of DNIS digits
  exit              Exit from cas custom mode
  invert-abcd       invert the ABCD bits before tx and after rx
  ka                KA Signal
  kd                KD Signal
  metering          R2 network is sending metering signal
  nc-congestion     Non Compelled Congestion signal
  no                Negate a command or set its defaults
  request-category  DNIS digits to be collected before requesting category
  unused-abcd       Unused ABCD bit values

Router(config-ctrl-cas)# country ?
  argentina             Argentina
  australia             Australia
  brazil                Brazil
  china                 China
  colombia              Colombia
  .
  .
  .
Router(config-ctrl-cas)# country argentina ?
  use-defaults   Use Country defaults
  <cr>

Router(config-ctrl-cas)# country argentina use-defaults
```

Note We highly recommend that you specify the default settings of your country. To display a list of supported countries, enter the **country?** command. The default setting for all countries is ITU.

R1 Modified Signaling Using an E1 Interface Example

The following example shows a configuration sample for R1 modified signaling on a Cisco access sever, using an E1 interface:

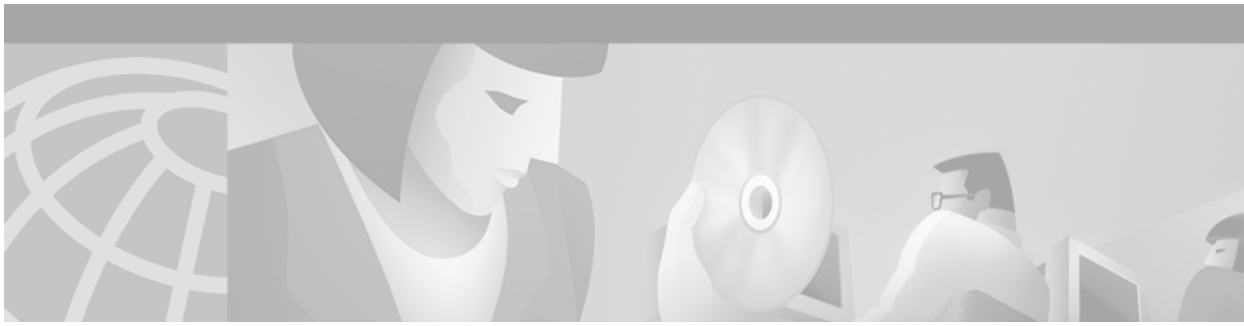
```
version xx.x
service timestamps debug datetime msec
no service password-encryption
!
hostname router
!
enable secret 5 $1$YAaG$L0jTcQ.nMH.gpFYXaOU5c.
!
no modem fast-answer
ip host dirt 10.255.254.254
ip multicast rpf-check-interval 0
isdn switch-type primary-dms100
!
!
controller E1 0
 clock source line primary
 cas-group 1 timeslots 1-15,17-31 type r1-modified  ani-dnis
!
controller E1 1
 clock source line secondary
 cas-group 1 timeslots 1-15,17-31 type r1-modified  ani-dnis
!
controller E1 2
 clock source internal
!
controller E1 3
 clock source internal
!
interface Ethernet0
 ip address 10.19.36.7 255.255.0.0
 no ip mroute-cache
!
interface FastEthernet0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
!
interface Group-Async1
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 480
 dialer-group 1
 async dynamic address
 async mode interactive
 peer default ip address pool DYNAMIC
 no fair-queue
 no cdp enable
 group-range 1 108
!
router igrp 200
 network 10.0.0.0
 network 192.168.254.0
!
no ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0
logging source-interface Ethernet0
```

```
!
line con 0
 exec-timeout 0 0
line 1 108
 exec-timeout 0 0
 modem InOut
 transport input all
line aux 0
line vty 0 4
!
end
```

R1 Modified Signaling for Taiwan Configuration Example

The following example shows how to configure R1 modified signaling for Taiwan:

```
service timestamps debug datetime msec
no service password-encryption
!
hostname router
!
enable secret 5 $1$YAaG$L0jTcQ.nMH.gpFYXaOU5c.
!
no modem fast-answer
ip host dirt 192.168.254.254
ip multicast rpf-check-interval 0
isdn switch-type primary-dms100
!
!
controller T1 1/1/0
 framing esf
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
 fdl att
!
controller T1 1/1/1
 framing esf
 linecode b8zs
 cablelength short 133
 cas-group 1 timeslots 1-24 type r1-modified
 fdl att
!
controller T1 1/1/2
 framing esf
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
 fdl att
!
controller T1 1/1/3
 framing esf
 linecode b8zs
 cablelength short 133
 pri-group timeslots 1-24
 fdl att
!
```



Configuring ISDN Special Signaling

This chapter describes features that either depend on special signaling services offered by an ISDN network service provider or overcome an inability to deliver certain signals. It describes these features in the following main sections:

- How to Configure ISDN Special Signaling
- Troubleshooting ISDN Special Signaling
- Configuration Examples for ISDN Special Signaling

For an overview of ISDN PRI, see the section "ISDN Service" in the "Overview of Dial Interfaces, Controllers, and Lines" chapter, and the section "ISDN Overview" in the "Configuring ISDN BRI" chapter.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the ISDN signaling commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

How to Configure ISDN Special Signaling

To configure special signaling features of ISDN, perform the tasks in the following sections; all tasks are optional:

- Configuring ISDN AOC (Optional)
- Configuring NFAS on PRI Groups (Optional)
- Enabling an ISDN PRI to Take PIAFS Calls on MICA Modems (Optional)
- Configuring Automatic Detection of Encapsulation Type (Optional)
- Configuring Encapsulation for Combinet Compatibility (Optional)

See the section "Configuration Examples for ISDN Special Signaling" at the end of this chapter for examples of these signaling features. See the "Troubleshooting ISDN Special Signaling" section later in this chapter for help in troubleshooting ISDN signaling features.

Configuring ISDN AOC

ISDN Advice of Charge (AOC) allows users to obtain charging information for all calls during the call (AOC-D) or at the end of the call (AOC-E) or both.

Users must have subscribed through their local ISDN network to receive the AOC information from the switch. No router configuration changes are required to retrieve this call charging information.

The ISDN AOC feature also supports, for the AOC-D service, an optional configurable short-hold mode that provides a dynamic idle timeout by measuring the call charging period, based on the frequency of the AOC-D or the AOC-E message from the network. The short-hold mode allows users to track call costs and to control and possibly reduce tariff charges. The short-hold mode idle time will do the following:

- Disconnect a call just before the beginning of a new charging period if the call has been idle for at least the configured minimum idle time.
- Maintain the call to the end of the current charging period past the configured idle timeout if the time left in the charging period is longer.

Incoming calls are disconnected using the static dialer idle timeout value.

The AOC-D and AOC-E messages are part of the Facility Information Element (IE) message. Its contents can be verified with the **debug q931** command. Call accounting information from AOC-D and AOC-E messages is stored in Simple Network Management Protocol (SNMP) MIB objects.

ISDN AOC is provided for ISDN PRI NET5 and ISDN BRI NET3 switch types only. AOC information at call setup is not supported.

Configuring Short-Hold Mode

No configuration is required to enable ISDN AOC. However, you can configure the optional short-hold minimum idle timeout period for outgoing calls; the default minimum idle timeout is 120 seconds. If the short-hold option is not configured, the router default is to use the static dialer idle timeout. If the short-hold idle timeout has been configured but no charging information is available from the network, the static dialer idle timeout applies.

To configure an ISDN interface and provide the AOC short-hold mode option on an ISDN interface, perform the following steps:

Step 1 Configure the ISDN BRI or PRI interface, as described in the chapter "Configuring ISDN BRI" or the section "How to Configure ISDN PRI" in the chapter "Configuring ISDN PRI" later in this publication, using the relevant keyword in the **isdn switch-type** command:

- BRI interface—**basic-net3**
- PRI interface—**primary-net5**

Step 2 Configure dialer profiles or legacy dial-on-demand routing (DDR) for outgoing calls, as described in the chapters in the "Dial-on-Demand Routing" part of this publication, making sure to do the following:

- Configure the static line-idle timeout to be used for incoming calls.
- For each destination, use the **dialer map** command with the **class** keyword (legacy DDR) or a **dialer string class** command (dialer profiles) to identify the dialer map class to be used for outgoing calls to the destination.

Step 3 Configure each specified dialer map class, providing a dialer idle timeout, or ISDN short-hold timeout, or both for outgoing calls, as described in this chapter.

To configure a dialer map class with timers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# map-class dialer` *classname*	Specifies the dialer map class and begins map class configuration mode.
Step 2	`Router(config-map-class)# dialer idle-timeout` *seconds*	(Optional) Specifies a static idle timeout for the map class to override the static line-idle timeout configured on the BRI interface.
Step 3	`Router(config-map-class)# dialer isdn short-hold` *seconds*	Specifies a dialer ISDN short-hold timeout for the map class.

Monitoring ISDN AOC Call Information

To monitor ISDN AOC call information, use the following command in EXEC mode:

Command	Purpose
`Router> show isdn {active [`*dsl* `|` *serial-number*`]	history [`*dsl* `|` *serial-number* `]	memory	nfas group` *group-number* `| service [`*dsl* `|` *serial-number*`]	status [`*dsl* `|` *serial-number*`]	timers [`*dsl* `|` *serial-number*`]}`	Displays information about active calls, call history, memory, nfas group, service or status of PRI channels, or Layer 2 or Layer 3 timers. The **history** keyword displays AOC charging time units used during the call and indicates whether the AOC information is provided during calls or at the end of calls. (The **service** keyword is available for PRI only.)

Configuring NFAS on PRI Groups

ISDN Non-Facility Associated Signaling (NFAS) allows a single D channel to control multiple PRI interfaces. A backup D channel can also be configured for use when the primary NFAS D channel fails.

Use of a single D channel to control multiple PRI interfaces can free one B channel on each interface to carry other traffic.

Any hard failure causes a switchover to the backup D channel and currently connected calls remain connected.

Once the channelized T1 controllers are configured for ISDN PRI, only the NFAS primary D channel must be configured; its configuration is distributed to all the members of the associated NFAS group.

ISDN NFAS Prerequisites

NFAS is only supported with a channelized T1 controller. Table 27 shows the Cisco IOS keywords for the ISDN switch types and lists whether NFAS is supported.

Table 27 ISDN Switch Types and NFAS Support

Switch Type	Keyword	NFAS Support
Lucent 4ESS Custom NFAS	**primary-4ess**	Yes
Lucent 5ESS Custom NFAS	**primary-5ess**	No (use National)
Nortel DMS Custom NFAS	**primary-dms**	Yes
NTT Custom NFAS	**primary-ntt**	Yes
National	**primary-ni**	Yes
Other switch types	—	No (use National)

Note On the Nortel (Northern Telecom) DMS-100 switch, when a single D channel is shared, multiple PRI interfaces may be configured in a single trunk group. The additional use of alternate route indexing, which is a feature of the DMS-100 switch, provides a rotary from one trunk group to another. This feature enables the capability of building large trunk groups in a public switched network.

The ISDN switch must be provisioned for NFAS. The primary and backup D channels should be configured on separate T1 controllers. The primary, backup, and B-channel members on the respective controllers should be the same as that configured on the router and ISDN switch. The interface ID assigned to the controllers must match that of the ISDN switch.

ISDN NFAS Configuration Task List

To configure NFAS on channelized T1 controllers configured for ISDN, perform the tasks in the following section: Configuring NFAS on PRI Groups (required).

You can also disable a channel or interface, if necessary, and monitor NFAS groups and ISDN service. To do so, perform the tasks in the following sections:

- Configuring NTT PRI NFAS (Optional)
- Disabling a Channel or Interface (Optional)
- Monitoring NFAS Groups (Optional)
- Monitoring ISDN Service (Optional)

See the section "NFAS Primary and Backup D Channels" later in this chapter for ISDN, NFAS, and DDR configuration examples.

Configuring NFAS on PRI Groups

This section documents tasks used to configure NFAS with D channel backup. When configuring NFAS, you use an extended version of the ISDN **pri-group** command to specify the following values for the associated channelized T1 controllers configured for ISDN:

- The range of PRI time slots to be under the control of the D channel (time slot 24).

- The function to be performed by time slot 24 (primary D channel, backup, or none); the latter specifies its use as a B channel.
- The group identifier number for the interface under control of the D channel.

To configure ISDN NFAS, use the following commands in controller configuration mode:

	Command	Purpose
Step 1	`Router(config-controller)# pri-group timeslots 1-24 nfas_d primary nfas_interface` *number* `nfas_group` *number*	On one channelized T1 controller, configures the NFAS primary D channel.
Step 2	`Router(config-controller)# pri-group timeslots 1-24 nfas_d backup nfas_interface` *number* `nfas_group` *number*	On a different channelized T1 controller, configures the NFAS backup D channel to be used if the primary D channel fails.
Step 3	`Router(config-controller)# pri-group timeslots 1-24 nfas_d none nfas_interface` *number* `nfas_group` *number*	(Optional) On other channelized T1 controllers, configures a 24-B-channel interface, if desired.

For an example of configuring three T1 controllers for the NFAS primary D channel, the backup D channel, and 24 B channels, along with the DDR configuration for the PRI interface, see the section "NFAS Primary and Backup D Channels" at the end of this chapter.

When a backup NFAS D channel is configured and the primary NFAS D channel fails, rollover to the backup D channel is automatic and all connected calls stay connected.

If the primary NFAS D channel recovers, the backup NFAS D channel remains active and does not switch over again unless the backup NFAS D channel fails.

Configuring NTT PRI NFAS

Addition of the NTT switch type to the NFAS feature allows its use in geographic areas where NTT switches are available. This feature provides use of a single D channel to control multiple PRI interfaces, and can free one B channel on each interface to carry other traffic.

To configure NTT PRI NFAS, use the procedure described in the "Configuring NFAS on PRI Groups" section. Specify a **primary-ntt** switch type.

Note You cannot configure a backup D channel for the NTT PRI NFAS feature; it does not support D channel backup.

Verifying NTT PRI NFAS

Step 1 Enter the **show isdn status** command to learn whether the ISDN PRI switch type was configured correctly:

```
Router# show isdn status serial 0:23

Global ISDN Switchtype = primary-ntt
ISDN Serial0:23 interface
```

Step 2 Enter the **show isdn nfas group** command to display information about members of an NFAS group:

```
Router# show isdn nfas group 1

ISDN NFAS GROUP 1 ENTRIES:
```

```
The primary D is Serial1/0:23.
The NFAS member is Serial2/0:23.

There are 3 total nfas members.
There are 93 total available B channels.
The primary D-channel is DSL 0 in state INITIALIZED.
The current active layer 2 DSL is 0.
```

Step 3 Enter the **show isdn service** command to display information about ISDN channels and the service states:

```
Router# show isdn service

PRI Channel Statistics:

ISDN Se1/0:23, Channel (1-24)
  Configured Isdn Interface (dsl) 0
  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 0 0 0 0 0 0 0 0 0 0 3
  Channel (1-24) Service (0=Inservice 1=Maint 2=Outofservice)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ISDN Se1/1:23, Channel (1-24)
  Configured Isdn Interface (dsl) 1
  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 0 0 0 0 0 0 0 0 0 0 0
  Channel (1-24) Service (0=Inservice 1=Maint 2=Outofservice)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ISDN Se2/0:23, Channel (1-24)
  Configured Isdn Interface (dsl) 2
  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 0 0 0 0 0 0 0 0 0 0 0
  Channel (1-24) Service (0=Inservice 1=Maint 2=Outofservice)
  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

Disabling a Channel or Interface

You can disable a specified channel or an entire PRI interface, thus taking it out of service or placing it into one of the other states that is passed in to the switch. To disable a specific channel or PRI interface, use one of the following commands in interface configuration mode as appropriate for your network:

Command	Purpose
`Router(config-if)# isdn service dsl` *number* `b_channel` *number* `state` *state-value*	Takes an individual B channel out of service or sets it to a different state.
`Router(config-if)# isdn service dsl` *number* `b_channel 0 state` *state-value*	Sets the entire PRI to the specified state.

The supported *state-values* are as follows:

- 0—In service
- 1—Maintenance
- 2—Out of service

When the T1 Controller Is Shut Down

In the event that a controller belonging to an NFAS group is shut down, all active B-channel calls on the controller that is shut down will be cleared (regardless of whether the controller is set to be primary, backup, or none), and one of the following events will occur:

- If the controller that is shut down is configured as the primary and no backup is configured, all active calls on the group are cleared.
- If the controller that is shut down is configured as the primary, and the active (In service) D channel is the primary and a backup is configured, then the active D channel changes to the backup controller.
- If the controller that is shut down is configured as the primary, and the active D channel is the backup, then the active D channel remains as backup controller.
- If the controller that is shut down is configured as the backup, and the active D channel is the backup, then the active D channel changes to the primary controller.

Note The active D channel changeover between primary and backup controllers happens only when one of the link fails and not when the link comes up. The T309 timer is triggered when the changeover takes place.

Monitoring NFAS Groups

To monitor NFAS groups, use the following command in EXEC mode:

Command	Purpose
`Router> show isdn nfas group number`	Displays information about members of an NFAS group.

Monitoring ISDN Service

To display information about ISDN channel service states, use the following command in EXEC mode:

Command	Purpose
`Router> show isdn service`	Displays information about ISDN channels and the service states.

Enabling an ISDN PRI to Take PIAFS Calls on MICA Modems

The Personal-Handyphone-System Internet Access Forum Standard (PIAFS) specifications describe a transmission system that uses the PHS 64000 bps/32000 bps unrestricted digital bearer on the Cisco AS5300 universal access server platform.

The PIAFS TA (terminal adapter) module is like a modem or a V.110 module in the following ways:

- Ports will be a pool of resources.
- Calls will use the same call setup Q.931 message.
- Module supports a subset of common AT commands.
- Call setup and teardown are similar.

However, the rate negotiation information will be part of the bearer cap and not the lower-layer compatibility. PIAFS calls will have the user rate as 32000 and 64000; this will be used to distinguish a PIAFS call from a V.110 call. Also, PIAFS will use only up to octets 5a in a call setup message. The data format will default to 8N1 for PIAFS calls.

To configure ISDN PRI to take PIAFS call on MICA modems, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial` *controller:channel*	Enters interface configuration mode for a D-channel serial interface.
Step 2	`Router(config-if)# isdn piafs-enabled`	Enables the PRI to take PIAFS calls on MICA modems.
Step 3	`Router(config-if)# exit`	Exits interface configuration mode.

Verifying PIAFS

Step 1 Enter the **show modem operational-status** *slot/port* command to view PIAFS call information.

```
Router# show modem op 1/32

Mdm Typ Status Tx/Rx G Duration RTS CTS DCD DTR
1/32 ISDN Conn 64000/64000 0 1d01h x x x x

Modem 1/32, Mica Hex Modem (Managed), Async33, tty33
Firmware Rev: 8.2.0.c
Modem config: Incoming and Outgoing
➔ Protocol: PIAFS, Compression: V.42bis both

Management config: Status polling
RX signals: 0 dBm

Last clearing of "show modem" counters never
2 incoming completes, 0 incoming failures
0 outgoing completes, 0 outgoing failures
0 failed dial attempts, 0 ring no answers, 0 busied outs
0 no dial tones, 0 dial timeouts, 0 watchdog timeouts
0 no carriers, 0 link failures, 0 resets, 0 recover oob
0 recover modem, 0 current fail count
0 protocol timeouts, 0 protocol errors, 0 lost events
0 ready poll timeouts
```

Configuring Automatic Detection of Encapsulation Type

You can enable a serial or ISDN interface to accept calls and dynamically change the encapsulation in effect on the interface when the remote device does not signal the call type. For example, if an ISDN call does not identify the call type in the lower-layer compatibility fields and is using an encapsulation that is different from the one configured on the interface, the interface can change its encapsulation type dynamically.

This feature enables interoperation with ISDN terminal adapters that use V.120 encapsulation but do not signal V.120 in the call setup message. An ISDN interface that by default answers a call as synchronous serial with PPP encapsulation can change its encapsulation and answer such calls.

Automatic detection is attempted for the first 10 seconds after the link is established or the first 5 packets exchanged over the link, whichever is first.

To enable automatic detection of encapsulation type, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# autodetect encapsulation` *encapsulation-type*	Enables automatic detection of encapsulation type on the specified interface.

You can specify one or more encapsulations to detect. Cisco IOS software currently supports automatic detection of PPP and V.120 encapsulations.

Configuring Encapsulation for Combinet Compatibility

Historically, Combinet devices supported only the Combinet Proprietary Protocol (CPP) for negotiating connections over ISDN B channels. To enable Cisco routers to communicate with those Combinet bridges, the Cisco IOS supports a the CPP encapsulation type.

To enable routers to communicate over ISDN interfaces with Combinet bridges that support only CPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation cpp`	Specifies CPP encapsulation.
Step 2	`Router(config-if)# cpp callback accept`	Enables CPP callback acceptance.
Step 3	`Router(config-if)# cpp authentication`	Enables CPP authentication.

Most Combinet devices support PPP. Cisco routers can communicate over ISDN with these devices by using PPP encapsulation, which supports both routing and fast switching.

Cisco 700 and 800 series routers and bridges (formerly Combinet devices) support only IP, Internet Protocol Exchange (IPX), and bridging. For AppleTalk, Cisco routers automatically perform half-bridging with Combinet devices. For more information about half-bridging, see the section "Configuring PPP Half-Bridging" in the chapter "Configuring Media-Independent PPP and Multilink PPP" later in this publication.

Cisco routers can also half-bridge IP and IPX with Combinet devices that support only CPP. To configure this feature, you only need to set up the addressing with the ISDN interface as part of the remote subnet; no additional commands are required.

Troubleshooting ISDN Special Signaling

To troubleshoot ISDN, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# debug dialer`	Displays the values of timers.

`Router# debug isdn q921 [interface bri number]`
 	
or
 	
`Router# debug isdn q921 interface serial slot/controller-number:23`
	Displays link layer information for all interfaces or, optionally, for a single BRI interface.
	
	Displays link layer information for a single PRI interface.

`Router# debug isdn q931 [interface bri number]`
 	
or
 	
`Router# debug isdn q931 interface serial slot/controller-number:23`
	Displays the content of call control messages and information elements, in particular the Facility IE message for all interfaces or, optionally, for a single BRI interface.
	
	Displays the content of call control messages and information elements, in particular the Facility IE message for a single PRI interface.

Configuration Examples for ISDN Special Signaling

This section provides the following configuration examples:

- ISDN AOC Configuration Examples
- ISDN NFAS Configuration Examples

ISDN AOC Configuration Examples

This section provides the following ISDN AOC configuration examples:

- Using Legacy DDR for ISDN PRI AOC Configuration
- Using Dialer Profiles for ISDN BRI AOC Configuration

Using Legacy DDR for ISDN PRI AOC Configuration

This example shows ISDN PRI configured on an E1 controller. Legacy DDR is configured on the ISDN D channel (serial interface 0:15) and propagates to all ISDN B channels. A static dialer idle-timeout is configured for all incoming calls on the B channels, but the map classes are configured independently of it. Map classes Kappa and Beta use AOC charging unit duration to calculate the timeout for the call. A short-hold idle timer is set so that if the line is idle for 10 or more seconds, the call is disconnected when the current charging period ends. Map class Iota uses a static idle timeout.

```
version 11.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname A
!
username c2503isdn password 7 1511021F0725
```

```
username B password 7 110A1016141D29
username C password 7 1511021F072508
isdn switch-type primary-net5
!
controller E1 0
 pri-group timeslots 1-31
!
interface Serial 0:15
 ip address 10.0.0.35 255.0.0.0
 encapsulation ppp
 dialer idle-timeout 150
 dialer map ip 10.0.0.33 name c2503isdn class Iota 06966600050
 dialer map ip 10.0.0.40 name B class Beta 778578
 dialer map ip 10.0.0.45 name C class Kappa 778579
 dialer-group 1
 ppp authentication chap
!
map-class dialer Kappa
 dialer idle-timeout 300
 dialer isdn short-hold 120
!
map-class dialer Iota
 dialer idle-timeout 300
!
map-class dialer Beta
 dialer idle-timeout 300
 dialer isdn short-hold 90
!
dialer-list 1 protocol ip permit
```

Using Dialer Profiles for ISDN BRI AOC Configuration

This example shows ISDN BRI configured as a member of two dialer pools for dialer profiles.

```
version 11.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname delorean
!
username spanky password 7 0705344245
username delorean password 7 1511021F0725
isdn switch-type basic-net3
!
interface BRI0
 description Connected to NTT 81012345678901
 no ip address
 dialer pool-member 1 max-link 1
 dialer pool-member 2 max-link
 encapsulation ppp
 no fair-queue
!
interface Dialer1
 ip address 10.1.1.8 255.255.255.0
 encapsulation ppp
 dialer remote-name spanky
 dialer string 81012345678902 class Omega
 dialer pool 1
 dialer-group 1
 ppp authentication chap
!
```

```
interface Dialer2
 ip address 10.1.1.8 255.255.255.0
 encapsulation ppp
 dialer remote-name dmsisdn
 dialer string 81012345678902 class Omega
 dialer string 14153909503 class Gamma
 dialer pool 2
 dialer-group 1
 ppp authentication chap
!
map-class dialer Omega
 dialer idle-timeout 60
 dialer isdn short-hold 150
!
map-class dialer Gamma
 dialer isdn short-hold 60
!
dialer-list 1 protocol ip permit
```

ISDN NFAS Configuration Examples

This section provides the following configuration examples:

- NFAS Primary and Backup D Channels
- PRI Interface Service State
- NTT PRI NFAS Primary D Channel Example

NFAS Primary and Backup D Channels

The following example configures ISDN PRI and NFAS on three T1 controllers of a Cisco 7500 series router. The NFAS primary D channel is configured on the 1/0 controller, and the NFAS backup D channel is configured on the 1/1 controller. No NFAS D channel is configured on the 2/0 controller; it is configured for 24 B channels. Once the NFAS primary D channel is configured, it is the only interface you see and need to configure; DDR configuration for the primary D channel—which is distributed to all B channels—is also included in this example.

```
isdn switch-type primary-4ess
!
! NFAS primary D channel on the channelized T1 controller in 1/0.
controller t1 1/0
 framing esf
linecode b8zs
 pri-group timeslots 1-24 nfas_d primary nfas_interface 0 nfas_group 1
!
! NFAS backup D channel on the channelized T1 controller in 1/1.
controller t1 1/1
 framing esf
linecode b8zs
 pri-group timeslots 1-24 nfas_d backup nfas_interface 1 nfas_group 1
!
! NFAS 24 B channels on the channelized T1 controller in 2/0.
controller t1 2/0
 framing esf
linecode b8zs
 pri-group timeslots 1-24 nfas_d none nfas_interface 2 nfas_group 1
!
```

```
! NFAS primary D channel interface configuration for PPP and DDR. This
! configuration is distributed to all the B channels in NFAS group 1 on the
! three channelized T1 controllers.
!
interface Serial 1/0:23
 ip address 10.1.1.2 255.255.255.0
 no ip mroute-cache
 encapsulation ppp
 dialer map ip 10.1.1.1 name flyboy 567898
 dialer map ip 10.1.1.3 name flyboy 101112345678
 dialer map ip 10.1.1.4 name flyboy 01112345678
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap
```

PRI Interface Service State

The following example puts the entire PRI interface back in service after it previously had been taken out of service:

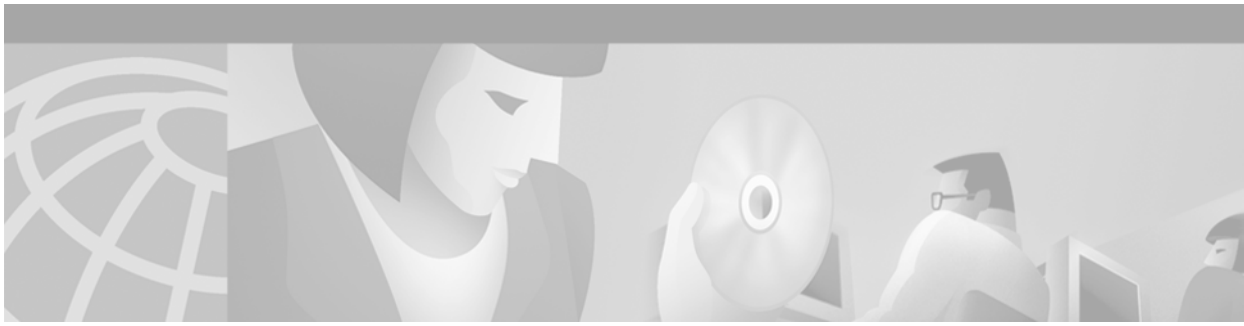
```
isdn service dsl 0 b-channel 0 state 0
```

NTT PRI NFAS Primary D Channel Example

The following example configures ISDN PRI and NFAS on three T1 controllers of a Cisco 7500 series router. The NFAS primary D channel is configured on the 1/0 controller. No NFAS D channel is configured on the 1/1 and 2/0 controllers; they are configured for 24 B channels. Once the NFAS primary D channel is configured, it is the only interface you see and need to configure. DDR configuration for the primary D channel—which is distributed to all B channels—is also included in this example.

```
isdn switch-type primary-ntt
!
! NFAS primary D channel on the channelized T1 controller in 1/0.
controller t1 1/0
 framing esf
linecode b8zs
 pri-group timeslots 1-24 nfas_d primary nfas_interface 0 nfas_group 1
!
! NFAS backup D channel on the channelized T1 controller in 1/1.
controller t1 1/1
 framing esf
linecode b8zs
 pri-group timeslots 1-24 nfas_d none nfas_interface 1 nfas_group 1
!
! NFAS 24 B channels on the channelized T1 controller in 2/0.
controller t1 2/0
 framing esf
linecode b8zs
 pri-group timeslots 1-24 nfas_d none nfas_interface 2 nfas_group 1
!
! NFAS primary D channel interface configuration for PPP and DDR. This
! configuration is distributed to all the B channels in NFAS group 1 on the
! three channelized T1 controllers.
!
interface Serial 1/0:23
 ip address 10.1.1.2 255.255.255.0
 no ip mroute-cache
 encapsulation ppp
 dialer map ip 10.1.1.1 name flyboy 567898
 dialer map ip 10.1.1.3 name flyboy 101112345678
```

```
dialer map ip 10.1.1.4 name flyboy 01112345678
dialer-group 1
no fair-queue
no cdp enable
ppp authentication chap
```



Configuring Network Side ISDN PRI Signaling, Trunking, and Switching

This chapter describes the Network Side ISDN PRI Signaling, Trunking, and Switching feature. The following main sections are provided:

- Network Side ISDN PRI Signaling Overview
- How to Configure Network Side ISDN PRI
- Configuration Examples for Network Side ISDN PRI Signaling, Trunking, and Switching

For hardware technical descriptions and for information about installing the controllers and interfaces, refer to the hardware installation and maintenance publication for your particular product.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the ISDN PRI commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Network Side ISDN PRI Signaling Overview

The Network Side ISDN PRI Signaling, Trunking, and Switching feature enables Cisco IOS software to replicate the public switched network interface to a PBX that is compatible with the National ISDN (NI) switch types and European Telecommunications Standards Institute (ETSI) Net5 switch types.

Routers and PBXs are both traditionally customer premises equipment (CPE) devices with respect to the public switched network interfaces. However, for Voice over IP (VoIP) applications, it is desirable to interface access servers to PBXs with the access server representing the public switched network.

Enterprise organizations use the current VoIP features with Cisco products as a method to reduce costs for long distance phone calls within and outside their organizations. However, there are times that a call cannot go over VoIP and the call needs to be placed using the Public Switched Telephone Network (PSTN). The customer then must have two devices connected to a PBX to allow some calls to be placed using VoIP and some calls to be placed over the PSTN. In contrast, this feature allows Cisco access servers to connect directly to user-side CPE devices such as PBXs and allows voice calls and data calls to be placed without requiring two different devices to be connected to the PBXs.

The Network Side ISDN PRI Signaling, Trunking, and Switching feature provides the following benefits:

- Allows you to bypass PSTN tariffed services such as trunking and administration, thus extending the cost savings of VoIP.
- Allows your PBXs to be connected directly to a Cisco access server, so PBX station calls can be routed automatically to the IP network without the need for special IP telephones.
- Provides flexibility in network design.
- Enables you to block calls selectively based on the called number or the calling number.

Call Switching Using Dial Peers

Call switching using dial peers enables Cisco VoIP gateways to switch both voice and data calls between different interfaces based on the dial peer matching. An incoming call is matched against configured dial peers, and based on the configured called number, the outgoing interface is selected. Any call that arrives from an ISDN PRI network side on a supported platform is either terminated on the access server, switched to an IP network, or switched to the PSTN, depending on the configuration.

Note An incoming call will be switched or processed as a voice call only if it matches a dial peer.

A dial peer is an addressable call endpoint identified, for example, by a phone number or a port number. In VoIP, there are two kinds of dial peers: plain old telephone service (POTS) and VoIP. Dial peers are defined from the perspective of the access server and are used for both inbound and outbound call legs. An *inbound* call leg originates outside the access server. An *outbound* call leg originates from the access server.

For inbound call legs, a dial peer might be associated with the calling number or the port designation. Outbound call legs always have a dial peer associated with them. The destination pattern (a defined initial part of a phone number) is used to identify the outbound dial peer. The call is associated with the outbound dial peer at setup time.

POTS dial peers associate a telephone number with a particular voice port so that incoming calls for that telephone number can be received and outgoing calls can be placed.

Additional information about dial peers can be found in the chapter “Configuring Dial Plans, Dial Peers, and Digit Manipulation” in the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.2.

Trunk Group Resource Manager

The Trunk Group Resource Manager (TGRM) supports the logical grouping, configuration, and joint management of one or more PRI interfaces. The TGRM is used to store configuration information and to accept or select an interface from a trunk group when requested. A trunk group is provisioned as the target of a dial peer, and the TGRM transparently selects the specific PRI interface and channels to use for incoming or outgoing calls. Trunks are selected based on usage: The trunk that is least used is selected.

Using trunk groups simplifies the task of configuring dial peers and PRI interfaces, and also enables the dynamic selection of PRI interfaces as needed in the access server.

A trunk group can include any number of PRI interfaces, but all the interfaces in a trunk group must use the same type of signaling.

Class of Restrictions

The class of restrictions (COR) functionality 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.

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.

ISDN Disconnect Timers

A new disconnect timer, T306, has been added as part of the Internetworking Signaling Enhancements for H.323 and SIP VoIP feature. This timer allows in-band announcements and tones to be played before a call is disconnected. It is designed for routers that are configured as an ISDN network-side switch. The T306 timer starts when the gateway receives a Disconnect message with a progress indicator of 8. The voice path is cut-through in the backward direction, and the announcement or error tone is played until the timer expires. When the timer expires, the voice application disconnects the call. You can configure this timer by using the **isdn t306** command. The T306 timer is supported only on routers that are configured for network-side ISDN. The following switches support network-side ISDN:

- National ISDN
- NET3 BRI
- NET5
- QSIG

The T310 timer sets a limit for a call in the Call Proceeding state. The timer starts when the router receives a Call Proceeding message and stops when the call moves to another phase, typically Alerting, Connect, or Progress. If the timer expires while the call is in the Call Proceeding state, the router releases the call. You can configure this timer by using the **isdn t310** command.

How to Configure Network Side ISDN PRI

See the following sections for configuration tasks for the Network Side ISDN PRI Signaling, Trunking, and Switching feature. Each task is identified as required or optional.

- Configuring ISDN Network Side (Required)
- Configuring Global or Interface Trunk Groups (Optional)
- Configuring Classes of Restrictions (Optional)
- Configuring ISDN T306 and T310 Timers (Optional)
- Verifying Network Side ISDN PRI Signaling, Trunking, and Switching (Optional)

The sections "Monitoring Network Side ISDN PRI" and "Monitoring TGRM" list commands that you can use to monitor network side ISDN PRI signaling.

Configuring ISDN Network Side

Before you begin to configure the Network Side ISDN PRI Signaling, Trunking, and Switching feature, ensure that the selected access server is in the following condition:

- The T1 or E1 controllers are operational and configured for ISDN PRI.
- The D-channel interfaces are operational and configured for ISDN PRI.
- Each D-channel interface is configured with the **isdn incoming-voice modem** command.

For example, the selected PRI interfaces might have a configuration similar to the following:

```
interface Serial1/0/0:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice modem
 no cdp enable
```

Also keep the following restrictions in mind as you configure network side ISDN PRI signaling, trunking, and switching:

- You can configure Cisco access server and access routers for either Network Side ISDN PRI for NI or Net5 switches.
- The trunking and COR parts of the Network Side ISDN PRI Signaling, Trunking, and Switching feature are available only on the Cisco AS5800 access server. In addition, call hairpinning without the need of a Voice Feature Card (and its digital signal processor) is available only on the Cisco AS5800 and Cisco AS5400. The remainder of the feature is platform-independent.
- The Cisco AS5800 and Cisco AS5400 switch both voice and data calls. The Cisco As5300 switches only data calls.
- On the Cisco AS5800, direct-inward-dial (DID) switched calls can work without a Voice Feature Card, if the appropriate modem is present. Refer to the AS5800 hardware and software installation manuals for more information.
- On the Cisco AS5400, direct-inward-dial (DID) switched calls can work with only Trunk Feature Cards present. No Voice Feature Card or Modem Feature card are required.
- An interface that is a member of a Non-Facility Associated Signaling (NFAS) group cannot belong to a trunk group.
- The Cisco AS5400 supports Network Side ISDN PRI Signaling and Calling Switching Using Dial Peers. It does not support Trunk Group Resource Manager and Class of Restrictions.
- The Network Side ISDN PRI part of this feature runs on any ISDN-capable platform with PRI interfaces. The trunking and class of restrictions parts of this feature require the Cisco AS5800.

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

Configuring ISDN Network Side for the National ISDN Switch Type

To configure Network Side ISDN PRI, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type type`	Sets the global ISDN switch type. Two types are supported: • **primary-ni** for NI on a T1 line • **primary-net5** for ETSI Net5 on an E1 line
	or	
	`Router(config-if)# interface serial0/0/n`	Specifies the D-channel interface. For *n*, the D-channel number, use: **0:23** on a T1 PRI **0:15** on an E1 PRI
	and	
	`Router(config-if)# switch-type primary-ni`	Sets the switch type on the interface.
Step 2	`Router(config-if)# isdn protocol-emulate network`	Enables network-side support on the PRI interface.

If you choose to configure Network Side ISDN PRI on individual interfaces in Step 1, repeat the configuration on the additional PRI interfaces.

Configuring ISDN Network Side for ETSI Net5 PRI

To configure a Cisco access router for ISDN Network Side for ETSI Net5 PRI, you can configure the **primary-net5** switch type globally or you can configure the **primary-net5** switch type on selected PRI interfaces. To configure ISDN Network Side for Net5, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type primary-net5`	Sets the **primary-net5** global ISDN switch type.
	or	or
	`Router(config-if)# interface serial0/0/0:15`	Specifies a D-channel interface to configure for ISDN Network Side for ETSI Net5 PRI.
	`Router(config-if)# switch-type primary-net5`	Sets the **primary-net5** switch type on the interface.
Step 2	`Router(config-if)# isdn protocol-emulate network`	Enables network side support on the interface.

Repeat the configuration steps on all the additional PRI D-channel interfaces you want to configure for ISDN Network Side for ETSI Net5 PRI.

Configuring Global or Interface Trunk Groups

You can create trunk groups globally (using the one-command version of Step 1) or on each interface (using the two-command version of Step 1). To configure trunk groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# trunk group group-number`	Defines the trunk group globally.
	or	
	`Router(config-if)# interface serial0/0/n`	Specifies the PRI D-channel. For *n*, the D-channel number, use: • **0:23** on a T1 PRI • **0:15** on an E1 PRI
	and	
	`Router(config-if)# trunk-group group-number`	Adds the interface to a trunk group. If the trunk group has not been defined globally, it will be created now.
Step 2	`Router(config-if)# max-calls {voice | data | any} number | [direction in | out]`	Applies a maximum number of calls restriction to the trunk group. This command can be repeated to apply a maximum number to different types of calls and, optionally, to specify whether the maximum applies to incoming or outgoing calls. **Note** Repeat Step 1 and Step 2 to create additional trunk groups and specify their restrictions, as needed for your traffic.
Step 3	`Router(config)# dial-peer voice tag pots`	Enters dial-peer configuration mode and defines a remote dial peer.
Step 4	`Router(config-dial-peer)# trunkgroup group-number`	Specifies the trunk group to be used for outgoing calls to the destination phone number.

Configuring Classes of Restrictions

To configure COR for dial peers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dial-peer cor custom`	Specifies that named classes of restrictions apply to dial peers and changes the command mode to COR configuration.
Step 2	`Router(config-cor)# name class-name`	Provides a name for a custom class of restrictions. **Note** Repeat this step for additional class names, as needed. These class names are used in various combinations to define the lists in Step 3 and Step 4.
Step 3	`Router(config)# dial-peer cor list list-name`	Provides a name for a list of restrictions.
Step 4	`Router(config-cor)# member class-name`	Adds a COR class to this list of restrictions. The member is a class named in Step 2. **Note** Repeat Step 3 and Step 4 to define another list and its membership, as needed.
Step 5	`Router(config)# dial-peer voice tag pots`	Enters dial-peer configuration mode and defines a remote dial peer.
Step 6	`Router(config-dial-peer)# corlist incoming cor-list-name`	Specifies the COR list to be used when this is the incoming dial peer.
Step 7	`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 5 through Step 7 for additional dial peers, as needed.

Configuring ISDN T306 and T310 Timers

To configure the T306 and T310 timers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial controller:timeslot`	Enters interface configuration mode for a D-channel serial interface.
Step 2	`Router(config-if)# isdn t306 milliseconds`	Sets the number of milliseconds that the gateway waits before clearing a call after it receives a Disconnect message with a progress indicator of 8.
Step 3	`Router(config-if)# isdn t310 milliseconds`	Sets the number of milliseconds that the gateway waits before clearing a call after it receives a Call Proceeding message.

To verify that the T306 timer is configured and operating correctly, perform the following steps:

Step 1 Display the running configuration file with the **show running-config** privileged EXEC command. Verify that the configuration is accurate for the T306 timer. See the "T306/T310 Timer Configuration Example" section for a sample configuration.

Step 2 Enable the **debug isdn q931** privileged EXEC command to trace the ISDN messages.

Step 3 Place a call to the gateway. Disconnect the call and allow the far end to play its error message until the T306 timer expires. When the timer expires, the gateway should disconnect the call.

Verifying Network Side ISDN PRI Signaling, Trunking, and Switching

To learn whether the Network Side ISDN PRI Signaling, Trunking, and Switching feature is configured successfully, perform the following steps:

Step 1 Enter the **show isdn status** command to learn whether an appropriate switch type is specified either globally or on the D-channel interface:

```
Router# show isdn status serial 0:15

Global ISDN Switchtype = primary-net5
ISDN Serial0:15 interface
        ******* Network side configuration *******
        dsl 0, interface ISDN Switchtype = primary-net5
```

Step 2 Enter the **show dial-peer voice** command to learn whether the trunk group COR list and permission fields are set as desired on a dial peer:

```
Router# show dial-peer voice

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:listA
        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,
```

Note The above output is for a dial peer configured with incoming COR list "listA" and without an outgoing COR list configured. When no outgoing COR list is configured, the **show dial-peer voice** command displays "minimum requirement" in the outgoing COR list output. When no incoming COR list is configured, the **show dial-peer voice** command displays "maximum capability" in the incoming COR list output.

Step 3 Enter the **show dial-peer cor** command to display the COR names and lists you defined. For example, if you configured COR as shown in the following sample display, the **show dial-peer cor** command output reflects that configuration.

Sample Configuration

```
dial-peer cor custom
 name 900block
 name 800_call
 name Catchall
!
dial-peer cor list list1
 member 900block
 member 800_call
!
dial-peer cor list list2
 member 900block
!
dial-peer cor list list3
 member 900block
 member 800_call
 member Catchall
```

Verification

```
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
```

Step 4 Enter the **show tgrm** command to verify the trunk group configuration. For example, if you configured trunk groups as shown in the following sample display, the **show tgrm** command output reflects that configuration.

Sample Configuration

```
interface Serial1/0/8:15
 no ip address
 ip mroute-cache
 no keepalive
 isdn switch-type primary-net5
 isdn protocol-emulate network
 isdn incoming-voice modem
 trunk-group 2
 no cdp enable
```

Verification

```
Router# show tgrm

      Trunk   Any in  Vce in   Data in
     Group #  Any out Vce out  Data out

        2       65535   65535   65535
                65535   65535   65535
                0 Retries
                Interface Se1/0/1:15    Data = 0, Voice = 0, Free = 30
                Interface Se1/0/8:15    Data = 2, Voice = 0, Free = 28

Total calls for trunk group:Data = 2, Voice = 0, Free = 58
                Selected Voice Interface :Se1/0/1:15
                Selected Data Interface  :Se1/0/1:15
```

Step 5 Enter the **show isdn status** command to display the status of both Network Side ISDN PRI and call switching:

```
Router# show isdn status

Global ISDN Switchtype = primary-net5
ISDN Serial1/0/0:15 interface
        ******* Network side configuration *******
        dsl 0, interface ISDN Switchtype = primary-net5
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Layer 3 Status:
        2 Active Layer 3 Call(s)
    Activated dsl 0 CCBs = 2
        CCB:callid=3C71, sapi=0, ces=0, B-chan=31, calltype=data
        CCB:callid=3C72, sapi=0, ces=0, B-chan=30, calltype=data
    The Free Channel Mask: 0x9FFF7FFF
ISDN Serial1/0/1:15 interface
/1/0/8
filtering...
ISDN Serial1/0/8:15 interface
        ******* Network side configuration *******
        dsl 8, interface ISDN Switchtype = primary-net5
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 0, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
    Layer 3 Status:
        2 Active Layer 3 Call(s)
    Activated dsl 8 CCBs = 2
        CCB:callid=BB40, sapi=0, ces=0, B-chan=1, calltype=DATA
        CCB:callid=BB41, sapi=0, ces=0, B-chan=2, calltype=DATA
    The Free Channel Mask: 0xFFFF7FFC
```

Monitoring Network Side ISDN PRI

To monitor Network Side ISDN PRI, use the following commands in EXEC mode as needed:

Command	Purpose
Router# **show controllers e1** *slot*/*port*	Checks Layer 1 (physical layer) of the PRI over E1.
Router# **show controllers e1** *number* **call-counters**	Displays the number of calls and call durations on an E1 controller.
Router# **show interfaces serial** *slot*/*port* **bchannel** *channel-number*	Displays information about the physical attributes of the ISDN PRI over channelized E1 B and D channels.
Router# **show isdn** {**active** | **history** | **memory** | **services** | **status** [*dsl* | *interface-type number*] | **timers**}	Displays information about memory, Layer 2 and Layer 3 timers, and the status of PRI channels.

Monitoring TGRM

To monitor and maintain the Trunk Group Resource Manager, use the following command in EXEC mode:

Command	Purpose
`Router# show tgrm`	Displays TGRM information for debugging purposes.

Configuration Examples for Network Side ISDN PRI Signaling, Trunking, and Switching

This section provides the following configuration examples:

- Call Switching and Dial Peers Configuration on T1/T3 Example
- Trunk Group Configuration Example
- COR for Dial Peer Configuration Example
- COR Based on Outgoing Dial Peers Example
- Dial Peers and Trunk Groups for Special Numbers Examples
- ISDN Network Side for ETSI Net5 PRI Configuration on E1 Example
- T306/T310 Timer Configuration Example

Call Switching and Dial Peers Configuration on T1/T3 Example

The following example enables Network Side ISDN PRI, call switching, and dial peers:

```
isdn switch-type primary-ni
!
controller T1 1/0/0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface Serial1/0/0:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice modem
 no cdp enable
!
dial-peer voice 11 pots
 incoming called-number 222
 destination-pattern 222
 direct-inward-dial
 port 1/0/0:D
 prefix 555
```

Trunk Group Configuration Example

The following trunk group allows only voice calls:

```
trunk group 1
max-calls data 0
!
```

The following trunk group allows a maximum of 20 outgoing voice calls:

```
trunk group 2
max-calls voice 20 direction out
!
```

The following trunk group allows a maximum of 50 incoming calls:

```
trunk group 3
max-calls any 50 direction in
!
```

The following trunk group allows a maximum of 100 calls, 30 of which can be voice (incoming or outgoing), and 60 of which can be incoming data (the remaining 10 will be unused):

```
trunk group 4
max-calls any 100
max-calls voice 30
max-calls data 60 direction in
```

COR for Dial Peer Configuration Example

The following example defines trunk group 101, establishes Network Side ISDN PRI on two PRI interfaces, and assigns both interfaces to trunk group 101. In addition, it establishes three COR lists, and specifies which incoming dial peers can make calls to 800 and which can make calls to 900 area codes. This example adopts a useful mnemonic pattern: the **dial-peer voice** tags for incoming calls correspond to the answer address (the phone number being called) and the **dial-peer voice** tags for outgoing calls correspond to the destination pattern.

```
trunk group 101
!
interface Serial1/0/0:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice modem
 no cdp enable
 trunk-group 101
!
interface Serial1/0/1:23
 no ip address
 no ip directed-broadcast
 isdn switch-type primary-ni
 isdn protocol-emulate network
 isdn incoming-voice modem
 no cdp enable
 trunk-group 101
!
dial-peer cor custom
 name 900_call
 name 800_call
!
dial-peer cor list list1
 member 900_call
!
```

```
dial-peer cor list list2
 member 800_call
!
dial-peer cor list list3
 member 900_csll
 member 800_call
!
dial-peer voice 525 pots
 answer-address 408525....
 corlist incoming list3
 direct-inward-dial
!
dial-peer voice 526 pots
 answer-address 408526....
 corlist incoming list2
 direct-inward-dial
!
dial-peer voice 900 pots
 destination-pattern 1900.......
 direct-inward-dial
 trunkgroup 101
 prefix 333
 corlist outgoing list1
!
dial-peer voice 12345 pots
 destination-pattern .T
 direct-inward-dial
 trunkgroup 202
!
```

COR Based on Outgoing Dial Peers Example

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 5x can have a **corlist outgoing** that contains COR 5x.

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.

```
config terminal
 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
```

In this example, calls from 525xxxx are not able to use dial peer 300, which means they will not be able to make 1900 calls (long distance calls to the 900 area code). But calls from 526xxxx can make 1900 calls.

Dial Peers and Trunk Groups for Special Numbers Examples

The following partial examples show setups for handling special numbers such as the 911 emergency number, the 0 local operator number, the 00 long-distance operator number, and so forth. “T” in these examples stands for the “interdigital timeout.” Calls to emergency numbers should not wait for this timeout, so 911 is used as the destination pattern, not 911T.

This partial example sets up a trunk group to handle calls going to the operator (0):

```
dial-peer voice 100 pots
 destination-pattern 0T
 trunkgroup 203
!
```

The following partial example sets up a trunk group to handle calls to the long distance operator (00):

```
dial-peer voice 200 pots
 destination-pattern 00T
 trunkgroup 205
!
```

The following partial example sets up a trunk group to handle calls to the international direct dial (011):

```
dial-peer voice 300 pots
 destination-pattern 011T
 trunkgroup 207
!
```

The following partial example sets up a trunk group to handle street line calls (calls that get a dial tone for an outside line):

```
disl-peer voice 400 pots
 destination-pattern 9T
 trunkgroup 209
!
```

The following partial example sets up a trunk group to handle calls for directory assistance:

```
dial-peer voice 500 pots
 destination-pattern 411
 trunkgroup 211
!
```

The following partial example sets up a trunk group to handle calls to the 911 emergency number. Emergency calls will not require a wait for the interdigital timeout to expire. They will be completed immediately.

```
dial-peer voice 600 pots
 destination pattern 911
 trunkgroup 333
```

ISDN Network Side for ETSI Net5 PRI Configuration on E1 Example

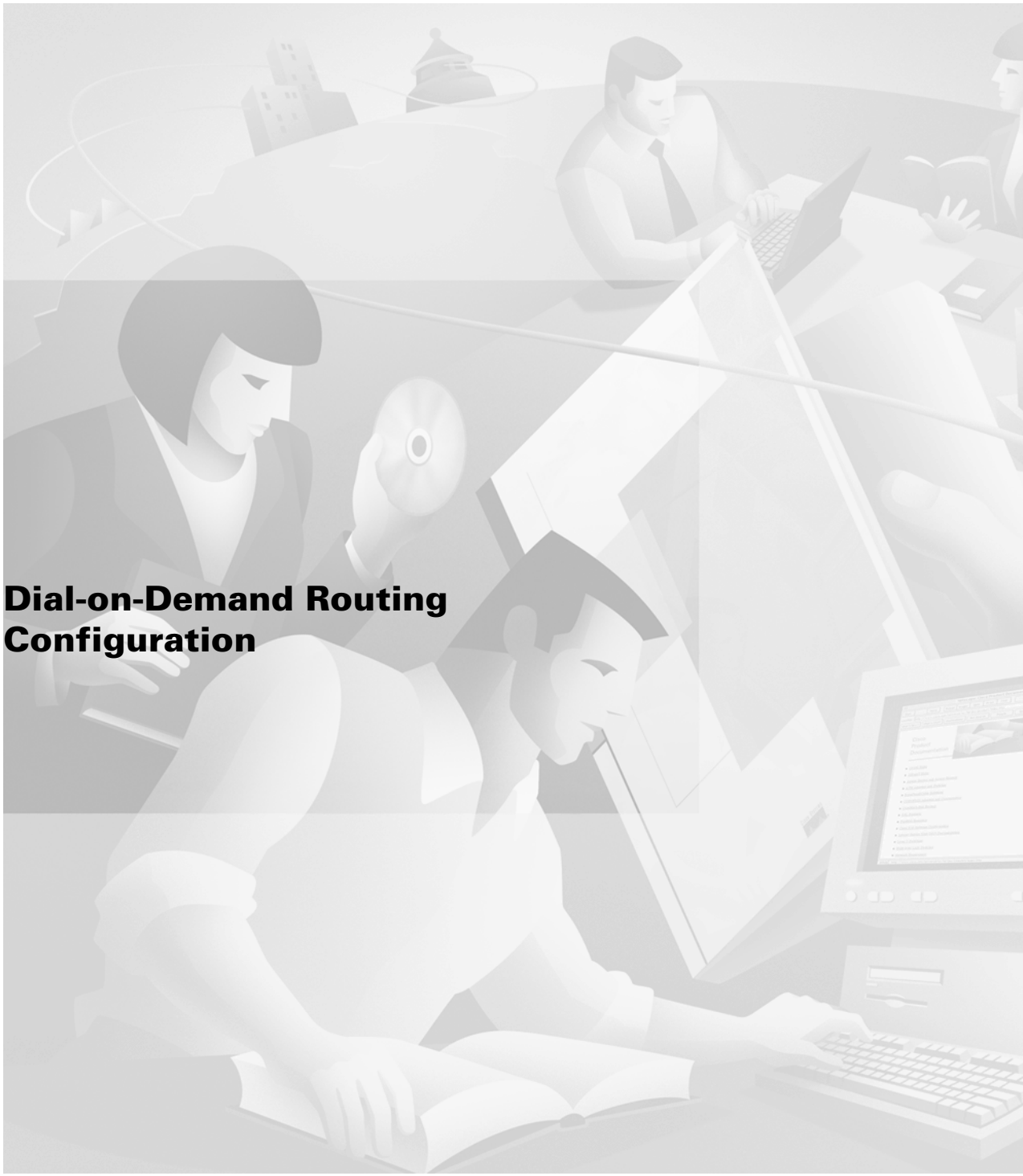
The following example enables the ISDN Network Side for ETSI Net5 PRI feature on an access server on which ISDN PRI is already configured and operational. In this example, the Net5 PRI switch type is set on the D-channel interface, and the global interface type is not shown.

```
controller e1 0
 pri-group timeslots 1-31
 exit
!
interface serial0:15
 no ip address
 no ip directed-broadcast
 ip mroute-cache
 isdn switch-type primary-net5
 isdn protocol-emulate networK
```

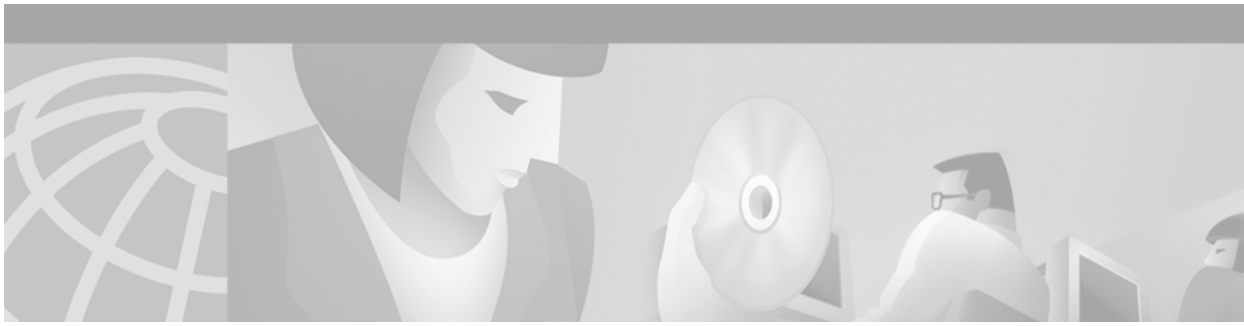
T306/T310 Timer Configuration Example

The following example configures the T306 and T310 disconnect timers:

```
interface Serial0:23
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer rotary-group 0
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 isdn t306 60000
 isdn t310 40000
```



Dial-on-Demand Routing Configuration



Preparing to Configure DDR

This chapter presents the decisions and preparations leading to a dial-on-demand routing (DDR) configuration and shows where some advanced features fit into the DDR configuration steps. It distinguishes between the topology decisions and the implementation of the decisions. In the implementation phase, it distinguishes the DDR-independent decisions from the DDR-dependent decisions.

This chapter provides the following information:

- DDR Decision Flowchart—A flowchart of topology and implementation decisions that you will need to make before you configure DDR.
- DDR Topology Decisions, DDR-Independent Implementation Decisions, and DDR-Dependent Implementation Decisions—References to sources of detailed information for the configuration steps associated with each decision.
- Global and Interface Preparations for DDR—Brief description indicating which preparations are global and which are interface-specific.
- Preparations for Routing or Bridging over DDR—A description of the steps required for bridging or routing over DDR.

The section "Configuration Examples for Legacy DDR" at the end of this chapter provides examples of configuring DDR in your network, and includes line configuration and chat script samples.

To identify the hardware platform or software image information associated with a feature, 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.

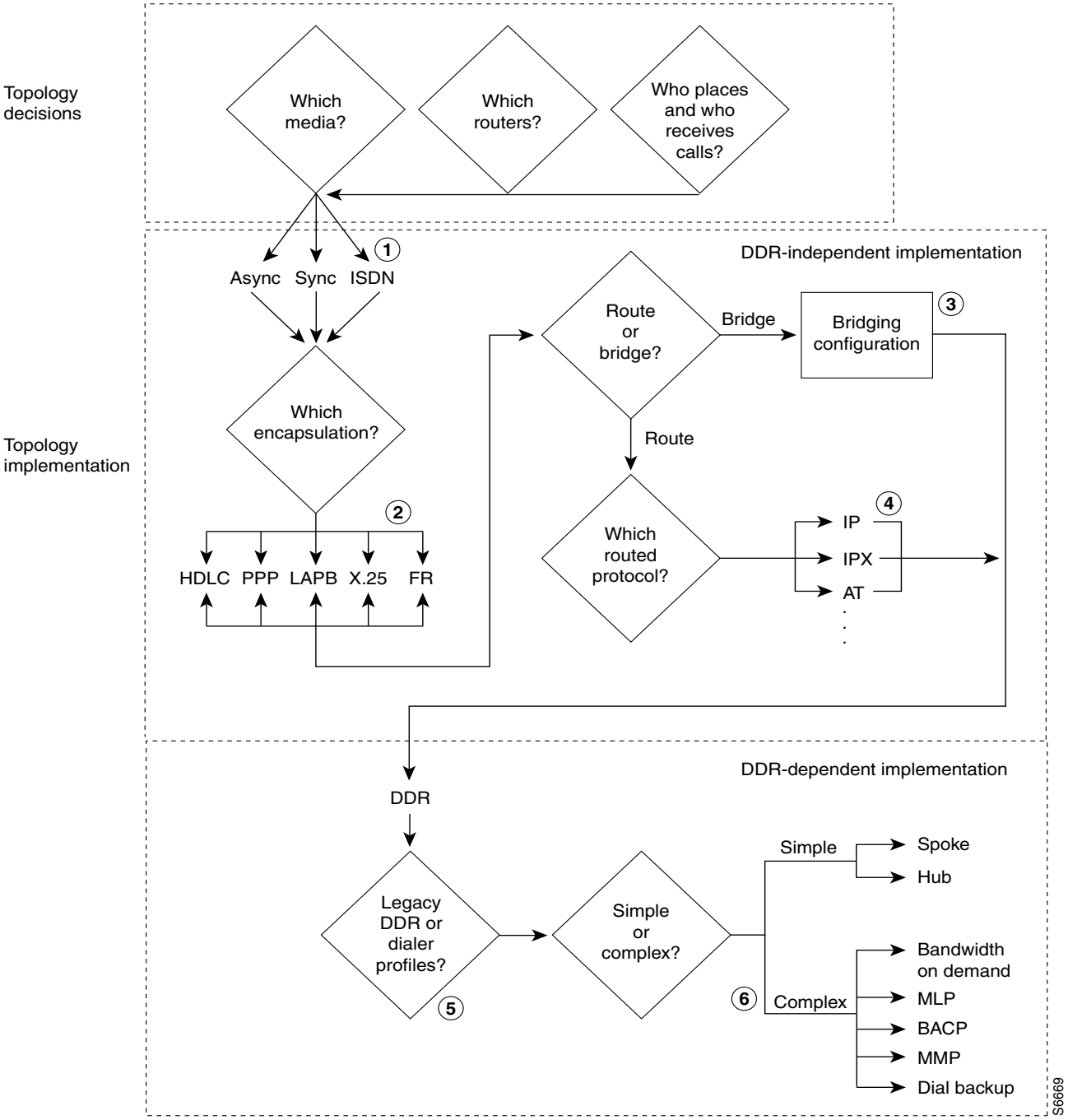
For a complete description of the global dialer commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

DDR Decision Flowchart

This section provides a flowchart of the decisions to be made before and while you configure DDR and also includes the flowchart.

Figure 48 presents the entire decision flowchart. The decision phases are shown in separate boxes. Numbers in parentheses refer to notes, which follow the figure.

Figure 48 Decisions and Implementation Flow to DDR



Flowchart Notes

The DDR chapters do not provide complete configuration information for most of the items in the following list. However, detailed information is available in other chapters and publications. The numbers in this list correspond to the circled numbers in the flowchart.

1. Configuration of the dial port and interface. The port, line, and interface are expected to be configured and operational before you configure DDR. See the relevant chapters in the "Preparing for Dial Access" part of this manual.
2. Encapsulation; including encapsulation for other WANs. See the "Configuring Media-Independent PPP and Multilink PPP" chapter of this publication for PPP encapsulation and refer to the *Cisco IOS Wide-Area Networking Configuration Guide* for sections on Frame Relay and X.25.
3. Bridging configurations. Refer to the *Cisco IOS Bridging and IBM Networking Configuration Guide*.
4. Routed protocols to be supported. See the protocol-specific chapters and publications.
5. Dialer profiles and legacy DDR are described in different chapters of the "Dial-on-Demand Routing" part of this publication.
6. Complex DDR configurations. Refer to the chapter "Configuring Media-Independent PPP and Multilink PPP" in this publication.

The DDR chapters provide complete configuration information about the simple hub-and-spoke DDR configurations, about the dialer profiles implementation of DDR, and about preparations required for configuring asynchronous interfaces for DDR.

DDR Topology Decisions

Topology decisions determine which routers will use DDR, which media and interfaces each one will use for DDR, and how each interface will function when using DDR. For example, if you choose a hub-and-spoke topology, one router will communicate with multiple routers. You must decide whether that router will use one interface or multiple interfaces for DDR, and whether it will receive calls only (forcing the spokes to initiate and bear the cost of calls). If it will use multiple interfaces, you must decide whether they will be of different types or the same type.

DDR-Independent Implementation Decisions

DDR-independent implementation decisions include the following:

- Using a specific interface or combination of interfaces for DDR.

  For complete configuration steps for the various media and interfaces, see the chapters in the "Dial-In Port Setup" part of this publication.

- Using nondefault encapsulations.

  The default encapsulation is High-Level Data Link Control (HDLC). However, PPP is widely used for situations in which authentication is desired, especially situations in which an interface will receive calls from multiple sites. Detailed PPP encapsulation requirements are described in the "Configuring Media-Independent PPP and Multilink PPP" and "Configuring Asynchronous PPP and SLIP" chapters of this publication.

If you decide to send DDR traffic over Frame Relay, X.25, or Link Access Procedure, Balanced (LAPB) networks, the interface must be configured with the appropriate encapsulation. For configuration details, refer to the related chapters in the Cisco *IOS Wide-Area Networking Configuration Guide*.

- Routing or bridging the DDR traffic.

  Legacy DDR supports bridging to only one destination, but the dialer profiles support bridging to multiple destinations.

  If you decide to bridge traffic over a dial-on-demand connection, configure the interface for transparent bridging. For detailed information, refer to the "Configuring Transparent Bridging" chapter of the *Cisco IOS Bridging and IBM Networking Configuration Guide*.

- Supporting one or more specific routed protocols, if you decide to route traffic.

  Depending on the protocol, you do need to control access by entering access lists and to decide how to support network addressing on an interface to be configured for DDR. You might also need to spoof keepalive or other packets. For configuration details, refer to the related network protocol chapters in the appropriate network protocols configuration guide, such as the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

DDR-Dependent Implementation Decisions

You must decide whether to implement legacy DDR or the newer dialer profiles; both are documented in the "Dial-on-Demand Routing" part of this publication. You must also decide whether a simple DDR configuration meets your business needs or whether to add other features.

Dialer Profiles

The dialer profiles implementation of DDR is based on a separation between logical and physical interface configuration. Dialer profiles also allow the logical and physical configurations to be bound together dynamically on a per-call basis.

Dialer profiles are advantageous in the following situations:

- When you want to share an interface (ISDN, asynchronous, or synchronous serial) to place or receive calls.
- When you want to change any configuration on a per-user basis.
- When you want to maximize ISDN channel usage using the Dynamic Multiple Encapsulations feature to configure various encapsulation types and per-user configurations on the same ISDN B channel at different times according to the type of call.
- When you want to bridge to many destinations, and for avoiding split horizon problem.

Most routed protocols are supported; however, International Organization for Standardization Connectionless Network Service (ISO CLNS) is not supported.

If you decide to configure dialer profiles, you must disable validation of source addresses for the routed protocols you support.

For detailed dialer profiles information, see the "Configuring Peer-to-Peer DDR with Dialer Profiles" chapter in this publication. For more information about Dynamic Multiple Encapsulations, see the "How to Configure Dialer Profiles" section in that chapter.

Legacy DDR

Legacy DDR is powerful and comprehensive, but its limitations affect scaling and extensibility. Legacy DDR is based on a static binding between the per-destination call specification and the physical interface configuration.

However, legacy DDR also has many strengths. It supports Frame Relay, ISO CLNS, LAPB, snapshot routing, and all routed protocols that are supported on Cisco routers. By default, legacy DDR supports fast switching.

For information about simple legacy DDR spoke configurations, see the "Configuring Legacy DDR Spokes" chapter. For information about simple legacy DDR hub configurations, see the "Configuring Legacy DDR Hubs" chapter. Both chapters are in this publication.

Simple or Complex DDR Configuration

You must also decide whether to implement a simple DDR configuration—whether it is a simple point-to-point (spoke-to-spoke) layout or a simple hub-and-spoke layout—or to add on features that make the implementation more complex. Add-on features include dial backup, bandwidth on demand, application of the Bandwidth Allocation Control Protocol (BACP), Multilink PPP, and many others.

Global and Interface Preparations for DDR

Some preparations are global and some depend on the type of interface you will configure for DDR. After you have made the required global decision whether to bridge or to route a specified protocol over a dial-on-demand link, you can make the following preparations:

- If you choose to bridge the protocol, decide whether to allow bridge packet access by Ethernet type codes or to permit all bridge packets across the link. Allowing access by Ethernet type codes requires you to define a bridging access list in global configuration mode.

  Allowing all bridge packets to trigger calls across a dial-on-demand link to a single destination is a DDR-dependent task addressed in the "Configure Dialer Access Lists to Trigger Outgoing Calls" section of both the "Configuring Legacy DDR Spokes" and "Configuring Legacy DDR Hubs" chapters in this publication.

  Bridging to multiple destinations requires dialer profiles.

- If you choose to route the protocol:
  - Define one or more access lists for the selected routed protocol to determine which packets should be permitted or denied access to the dial-on-demand link.

    Allowing those packets to trigger calls across a dial-on-demand link is a DDR-dependent task addressed in the "Configure Dialer Access Lists to Trigger Outgoing Calls" section of both the "Configuring Legacy DDR Spokes" and "Configuring Legacy DDR Hubs" chapters in this publication.
  - Define an appropriate dialer list for the protocol.
  - Disable validation of source addresses, if you decide to configure dialer profiles.

Preparations Depending on the Selected Interface Type

The steps shown in this chapter assume that you have also completed the required preparatory steps for the type of interface you will configure for DDR:

- The interface is installed, the cable is connected as needed, and operational.
- Chat scripts are ready, as needed, for any asynchronous interfaces and modem scripts have been assigned to the relevant asynchronous lines.
- Asynchronous lines and modems are configured and operational, as needed.
- Any ISDN line that will be used for DDR is properly provisioned and running.
- You have decided which interfaces and how many interfaces are to be configured for DDR, and what functions each interface will perform.

Preparations for Routing or Bridging over DDR

The following tasks are DDR-independent and can be completed before you configure DDR. Minimal tasks required for each item are presented in this chapter. For detailed information about bridging, routing, and wide-area networking configurations, refer to the appropriate chapters in other manuals of the Cisco IOS documentation set.

Complete the following minimal tasks for the global decisions you have made:

- Preparing for Transparent Bridging over DDR (As required)
- Preparing for Routing over DDR (As required)

Preparing for Transparent Bridging over DDR

To prepare for transparent bridging over DDR, complete the tasks in the following sections:

- Defining the Protocols to Bridge (As required)
- Specifying the Bridging Protocol (As required)
- Controlling Bridging Access (As required)

Defining the Protocols to Bridge

IP packets are routed by default unless they are explicitly bridged; all others are bridged by default unless they are explicitly routed. To bridge IP packets, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip routing`	Disables IP routing.

If you choose *not* to bridge another protocol supported on your network, use the relevant command to enable routing of that protocol. For more information about tasks and commands, refer to the relevant protocol chapter in the appropriate network protocols configuration guide, such as the *Cisco IOS AppleTalk and Novell IPX Configuration Guide* or *Cisco IOS IP Configuration Guide*.

Specifying the Bridging Protocol

You must specify the type of spanning-tree bridging protocol to use and also identify a bridge group. To specify the spanning-tree protocol and a bridge group number, use the following command in global configuration mode:

Command	Purpose
`Router(config)# bridge bridge-group protocol {ieee | dec}`	Defines the type of spanning tree protocol and identifies a bridge group.

The bridge-group number is used when you configure the interface and assign it to a bridge group. Packets are bridged only among members of the same bridge group.

Controlling Bridging Access

You can control access by defining any transparent bridge packet as *interesting*, or you can use the finer granularity of controlling access by Ethernet type codes.

To control access by Ethernet type codes, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number {permit | deny} type-code [mask]`	Identifies interesting packets by Ethernet type codes (access list numbers must be in the range 200–299).
Step 2	`Router(config)# dialer-list dialer-group protocol bridge list access-list-number`	Defines a dialer list for the specified access list.

Packets with a specified Ethernet type code can trigger outgoing calls. Spanning tree bridge protocol data units (BPDUs) are always treated as *uninteresting* and cannot trigger calls.

For a table of some common Ethernet types codes, refer to the "Ethernet Types Codes" appendix in the *Cisco IOS Bridging and IBM Networking Command Reference*.

To identify all transparent bridge packets as interesting, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer-list dialer-group protocol bridge permit`	Defines a dialer list that treats all transparent bridge packets as interesting.

Preparing for Routing over DDR

DDR supports the following routed protocols: AppleTalk, Banyan VINES, DECnet, IP, Internet Protocol Exchange (IPX), ISO CLNS, and Xerox Network Systems (XNS).

To prepare for routing a protocol over DDR, perform the tasks in the relevant section:

- Configuring the Protocol for Routing and Access Control (As required)
- Associating the Protocol Access List with a Dialer Group (As required)

Configuring the Protocol for Routing and Access Control

This section specifies the minimal steps required to configure a protocol for routing over DDR. For more options and more detailed descriptions, refer to the relevant protocol chapter.

Configuring IP Routing

IP routing is enabled by default on Cisco routers; thus no preparation is required simply to enable it. You might, however, need to decide your addressing strategy and complete other global preparations for routing IP in your networks. To use dynamic routing where multiple remote sites communicate with each other through a central site, you might need to disable the IP split horizon feature. Refer to the "Configuring IP Addressing" chapter in the *Cisco IOS IP Configuration Guide* for more information.

At a minimum, you must complete the following tasks:

- Disable validation of source addresses.
- Configure one or more IP access lists before you refer to the access lists in DDR **dialer-list** commands to specify which packets can trigger outgoing calls.

To disable validation of source addresses, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# router rip`	Specifies the routing protocol; RIP, for example.
`Router(config)# no validate-update-source`	Disables validation of source addresses.
`Router(config)# network number`	Specifies the IP address.

For more information about IP routing protocols, refer to the *Cisco IOS IP Configuration Guide*.

To configure IP access lists, use one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# access-list access-list-number {deny | permit} source [source-mask]`
 	
or
 	
`Router(config)# access-list access-list-number {deny | permit} protocol source source-mask destination destination-mask [operator operand]`
	Specifies an IP standard access list.
	
	Specifies an IP extended access list.

You can also use simplified IP access lists that use the **any** keyword instead of the numeric forms of source and destination addresses and masks. Other forms of IP access lists are also available. For more information, refer to the "IP Services Commands" chapter in the *Cisco IOS IP Configuration Guide*.

For an example of configuring DDR for IP, see the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" in this publication.

You can configure IP routing on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Configuring Novell IPX Routing

To configure routing of IPX over DDR, you must complete both global and interface-specific tasks:

- Enable IPX routing globally.
- Enable IPX watchdog spoofing, or enable Sequenced Packet Exchange (SPX) keepalive spoofing on the interface.

To enable IPX routing, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx routing [node]`	Enables IPX routing.

To enable IPX watchdog spoofing on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx watchdog-spoof`	Enables IPX watchdog spoofing.

To enable SPX keepalive spoofing, use the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# ipx spx-spoof`	Enables SPX keepalive spoofing.
`Router(config-if)# ipx spx-idle-time delay-in-seconds`	Sets the idle time after which SPX spoofing begins.

You can configure IPX routing on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

For detailed DDR for IPX configuration examples, refer to the section "IPX over DDR Example" in the "Configuring Novell IPX" chapter of the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Configuring AppleTalk Routing

You must enable AppleTalk routing and then specify AppleTalk access lists. After you specify AppleTalk access lists, define dialer lists. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword.

You can configure AppleTalk routing on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" for more information and examples.

Configuring Banyan VINES Routing

To configure DDR for Banyan VINES, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# vines access-list access-list-number {permit | deny} source source-mask1`	Specifies a VINES standard access list.
or	
`Router(config)# vines access-list access-list-number {permit | deny} source source-mask [destination] [destination-mask]`	Specifies a VINES extended access list.

After you specify VINES standard or extended access lists, define DDR dialer lists. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" for more information and examples.

You can configure Banyan VINES on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Note The Banyan VINES **neighbor** command is not supported for LAPB and X.25 encapsulations.

Configuring DECnet Routing

To configure DDR for DECnet, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {permit | deny} source source-mask1`	Specifies a DECnet standard access list.
or	
`Router(config)# access-list access-list-number {permit | deny} source source-mask [destination] [destination-mask]`	Specifies a DECnet extended access list.

After you specify DECnet standard or extended access lists, define DDR dialer lists. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" in this publication for more information and examples.

You classify DECnet control packets, including hello packets and routing updates, using one or more of the following commands: **dialer-list protocol decnet_router-L1 permit**, **dialer-list protocol decnet_router-L2 permit**, and **dialer-list protocol decnet_node permit**.

You can configure DECnet on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Configuring ISO CLNS Routing

To configure ISO CLNS for DDR, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# clns filter-set` *name* [**permit** | **deny**] *template*	Specifies one or more CLNS filters, repeating this command as needed to build the filter list associated with the filter name.
Step 2	`Router(config)# interface` *type number*	Specifies the interface to apply the filter to and begins interface configuration mode.
Step 3	`Router(config-if)# clns access-group` *name* **out**	Filters CLNS traffic going out of the interface, on the basis of the filter specified and named in Step 1.

After you complete these CLNS-specific steps, define a dialer list for CLNS. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. Use the *access-group* argument with this command, because ISO CLNS uses access groups but does not use access lists. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" in this publication for more information and examples.

You classify CLNS control packets, including hello packets and routing updates, using the **dialer-list protocol clns_is permit** and/or **dialer-list protocol clns_es permit** command.

You can configure ISO CLNS on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Configuring XNS Routing

You must enable XNS routing and then define an access list. To define an XNS access list, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list` *access-list-number* {**deny** | **permit**} *source-network*[**.***source-address* [*source-address-mask*]] [*destination-network*[**.***destination-address* [*destination-address-mask*]]]	Specifies a standard XNS access list.
or	
`Router(config)# access-list` *access-list-number* {**deny** | **permit**} *protocol* [*source-network*[**.***source-host* [*source-network-mask***.**]*source-host-mask*] *source-socket* [*destination-network* [**.***destination-host* [*destination-network-mask***.***destination-host-mask*] *destination-socket*[**/pep**]]]	Specifies an extended XNS access list.

After you specify an XNS access list, define a DDR dialer list. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the chapters "Configuring a Legacy DDR Spoke" or "Configuring a Legacy DDR Hub" for more information and examples.

You can configure XNS on DDR asynchronous, synchronous serial, and ISDN interfaces, as well as dialer rotary groups.

Associating the Protocol Access List with a Dialer Group

DDR supports the following routed protocols: AppleTalk, Banyan VINES, DECnet, IP, Novell IPX, ISO CLNS, and XNS.

You can permit or deny access by protocol, or you can specify an access list for more refined control. To associate a protocol or access list with a dialer group, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer-list` *dialer-group* `protocol` *protocol-name* {`permit` | `deny` | `list` *access-list-number* | *access-group*}	Associates a protocol access list number or access group name with the dialer group.

Note For a given protocol and a given dialer group, only one access list can be specified in the **dialer-list** command.

For the **dialer-list protocol list** command form, acceptable access list numbers are as follows:

- Banyan VINES, DECnet, IP, and XNS standard and extended access list numbers
- Novell IPX standard, extended, and SAP access list numbers
- AppleTalk access lists numbers
- Bridge type codes

Configuration Examples for Legacy DDR

The following sections provide DDR configuration examples:

- Point-to-Point DDR Without Authentication Examples
- Point-to-Point DDR with Authentication Examples

Point-to-Point DDR Without Authentication Examples

The following example sets up two-way reciprocal DDR without authentication; the client and server have dial-in access to each other. This configuration is demonstrated in the following two subsections.

Remote Configuration

The following sample configuration is performed on the remote side of the connection:

```
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 ip address 172.30.45.2 255.255.255.0
```

```
 async mode dedicated
 peer default ip address 172.30.45.1
 encapsulation ppp
 dialer in-band
 dialer string 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 async 7
 ip default-network 172.30.0.0
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
line 7
 no exec
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```

Local Configuration

The following sample configuration is performed on the local side of the connection:

```
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode dedicated
 peer default ip address 172.30.45.2
 encapsulation ppp
 dialer in-band
 dialer string 1235
 dialer rotary-group 1
!
interface async 8
 async mode dedicated
 peer default ip address 172.30.45.2
 dialer rotary-group 1
!
ip route 172.30.44.0 255.255.255.0 async 7
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name remote 4321
 dialer load-threshold 80
!
ip route 172.30.44.0 255.255.255.0 128.150.45.2
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
route igrp 109
network 172.30.0.0
redistribute static
passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```

Point-to-Point DDR with Authentication Examples

The following sample sets up two-way DDR with authentication; the client and server have dial-in access to each other. This configuration is demonstrated in the following two subsections.

Remote Configuration

The following example is performed on the remote side of the connection. It provides authentication by identifying a password that must be provided on each end of the connection.

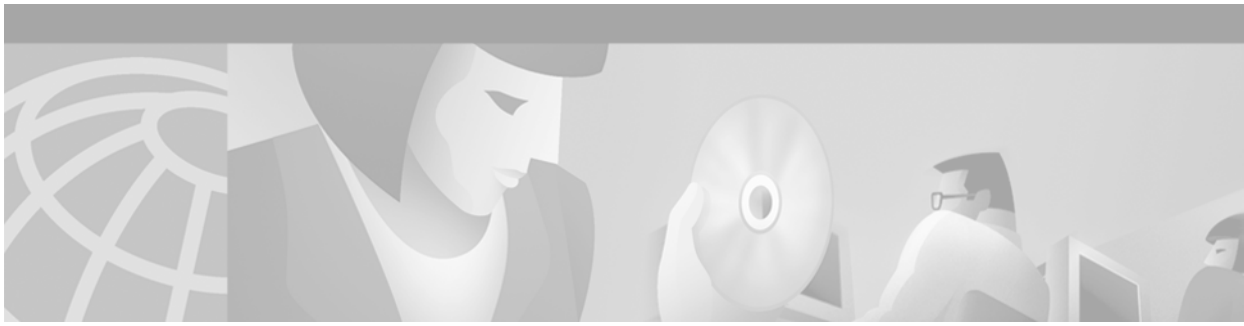
```
username local password secret1
username remote password secret2
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 ip address 172.30.45.2 255.255.255.0
 async mode dedicated
 peer default ip address 172.30.45.1
 encapsulation ppp
 dialer in-band
 dialer string 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 async 7
 ip default-network 172.30.0.0
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
line 7
 no exec
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```

Local Configuration

The following example configuration is performed on the local side of the connection. As with the remote side configuration, it provides authentication by identifying a password for each end of the connection.

```
username remote password secret1
username local password secret2
!
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode dedicated
 peer default ip address 172.30.45.2
 dialer rotary-group 1
!
interface async 8
 async mode dedicated
 peer default ip address 172.30.45.2
 dialer rotary-group 1
!
interface dialer 1
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
```

```
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name remote 4321
 dialer load-threshold 80
!
ip route 172.30.44.0 255.255.255.0 172.30.45.2
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
 route igrp 109
 network 172.30.0.0
 redistribute static
 passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 script dialer generic
```



Configuring Legacy DDR Spokes

This chapter describes how to configure legacy dial-on-demand routing (DDR) on interfaces that function as a *spoke* in a hub-and-spoke network topology. It includes the following main sections:

- DDR Spokes Configuration Task Flow
- How to Configure DDR
- Monitoring DDR Connections
- Configuration Examples for Legacy DDR Spoke

This chapter considers a spoke interface to be any interface that calls or receives calls from exactly one other router, and considers a hub interface to be an interface that calls or receives calls from more than one router: all the spokes in the network.

This chapter also describes the DDR-independent tasks required to bridge protocols or to route protocols over DDR. Most of these tasks are global in scope and can be completed before you begin to configure DDR.

For configuration tasks for the central hub interface in a hub-and-spoke network topology, see the chapter "Configuring a Legacy DDR Hub" in this publication.

For information about the Dialer Profiles implementation of DDR, see the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles" in this publication.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the legacy DDR spoke commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

DDR Spokes Configuration Task Flow

Before you configure DDR, make sure you have completed the preparations for bridging or routing as described in the chapter "Preparing to Configure DDR" in this publication. That chapter provides information about the minimal requirements. For detailed information about bridging, routing, and wide-area networking configurations, refer to the appropriate chapters in other volumes of this documentation set.

When you configure DDR on a spoke interface in a hub-and-spoke topology, you perform the following general steps:

Step 1 Specify the interface that will place calls to or receive calls from a single site. (See the chapter "Configuring Legacy DDR Hubs" in this publication for information about configuring an interface to place calls to or receive calls from multiple sites.)

Step 2 Enable DDR on the interface. This step is not required for some interfaces; for example, ISDN interfaces and passive interfaces that receive only from DTR-dialing interfaces.

Step 3 Configure the interface to receive calls only, if applicable. Receiving calls from multiple sites requires each inbound call to be authenticated.

Step 4 Configure the interface to place calls only, if applicable.

Step 5 Configure the interface to place and receive calls, if applicable.

Step 6 If the interface will place calls, specify access control for:

- Transparent bridging—Assign the interface to a bridge group, and define dialer lists associated with the bridging access lists. The interface switches between members of the same bridge group, and dialer lists specify which packets can trigger calls.

  or

- Routed protocols—Define dialer lists associated with the protocol access lists to specify which packets can trigger calls.

Step 7 Customize the interface settings (timers, interface priority, hold queues, bandwidth on demand, and disabling fast switching) as needed.

When you have configured the interface and it is operational, you can monitor its performance and its connections as described in the "Monitoring DDR Connections" section later in this chapter.

You can also enhance DDR by configuring Multilink PPP and configuring PPP callback. The PPP configuration tasks are described in the chapter "Configuring Media-Independent PPP and Multilink PPP" in this publication.

See the section "Configuration Examples for Legacy DDR Spoke" later in this chapter for examples of how to configure DDR on your network.

How to Configure DDR

To configure DDR on an interface, perform the tasks in the following sections. The first five bulleted items are required. The remaining tasks are not required, but might be necessary in your networking environment.

- Specifying the Interface (Required)
- Enabling DDR on the Interface (Required)
- Configuring the Interface to Place Calls (Required)
  or
  Configuring the Interface to Receive Calls (Required)
  or
  Configuring the Interface to Place and Receive Calls (Required)
- Defining the Traffic to Be Authenticated (As required)

- Configuring Access Control for Outgoing Calls (As required)
- Configuring Access Control for Bridging (As required)
- Configuring Access Control for Routing (As required)
- Customizing the Interface Settings (As required)
- Sending Traffic over Frame Relay, X.25, or LAPB Networks (As required)

You can also monitor DDR connections. See the "Monitoring DDR Connections" section later in this chapter for commands and other information.

For examples of legacy DDR on a point-to-point connection, see the "Configuration Examples for Legacy DDR Spoke" section later in this chapter.

Specifying the Interface

This section assumes that you have completed any preparatory steps required for the relevant interface. For example, if you intend to use an asynchronous interface, it assumes that you have completed the modem support and line configuration steps and the chat script creation steps. If you intend to use an ISDN interface, it assumes that you have the ISDN line properly provisioned and running.

You can configure any asynchronous, synchronous serial, ISDN, or dialer interface for legacy DDR.

Note When you specify an interface, make sure to use the interface numbering scheme supported on the network interface module or other port hardware on the router. On the Cisco 7200 series, for example, you specify an interface by indicating its type, slot number, and port number.

To specify an interface to configure for DDR, use one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# interface async number`
`Router(config)# interface serial number`
`Router(config)# interface bri number`
or
	Specifies an interface to configure for DDR.

`Router(config)# interface serial slot/port:23`	Specifies an ISDN PRI D channel (T1).
`Router(config)# interface serial slot/port:15` or	Specifies an ISDN PRI D channel (E1).
`Router(config)# interface dialer number`	Specifies a logical interface to function as a dialer rotary group leader.

Dialer interfaces are logical or virtual entities, but they use physical interfaces to place or receive calls.

Enabling DDR on the Interface

This task is required for asynchronous or synchronous serial interfaces but not for ISDN interfaces. The software automatically configures ISDN interfaces to be dialer type ISDN.

This step is not required for ISDN interfaces (BRI interfaces and ISDN PRI D channels) and for *purely passive* interfaces that will receive calls only from interfaces that use DTR dialing.

Enabling DDR on an interface usually requires you to specify the type of dialer to be used. This step is not required for ISDN interfaces because the software automatically configures ISDN interfaces to be dialer type ISDN.

To enable DDR and specify the dialer type, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# dialer dtr` or	Enables DDR and configures the specified serial interface to use DTR dialing—for interfaces with non-V.25*bis* modems using EIA Data Terminal Ready (DTR) signaling.
`Router(config)# dialer in-band [no-parity | odd-parity]`	Enables DDR and configures the specified serial interface to use in-band dialing—for asynchronous interfaces or interfaces using V.25*bis* modems.

Note An interface configured with the **dialer in-band** command can both place and receive calls. A serial interface configured for DTR dialing can place calls only; it cannot accept them.

You can optionally specify parity if the modem on this interface uses the V.25*bis* command set. The 1984 version of the V.25*bis* specification states that characters must have odd parity. However, the default for the **dialer in-band** command is no parity.

For an example of configuring an interface to support DTR dialing, see the section "DTR Dialing Example" later in this chapter.

To receive calls from an interface that is using DTR dialing, an interface can be configured for in-band dialing or not configured for anything but encapsulation, depending on the desired behavior. If you expect the receiving interface to terminate a call when no traffic is received for some time, you must configure in-band dialing (along with access lists and a dummy dialer string). If the receiving interface is purely passive, no additional configuration is necessary.

Note You can configure an interface or dialer rotary group to both place and receive calls. If the interface is calling and being called by a single site, simply enable DDR and specify a dial string.

Configuring the Interface to Place Calls

To configure an interface to place calls to one site only, perform the tasks in one of the following sections:

- Specifying the Dial String for Synchronous Serial Interfaces (As required)
- Specifying Chat Scripts and Dial Strings for Asynchronous Serial Interfaces (As required)

Specifying the Dial String for Synchronous Serial Interfaces

If you want to call only one remote system per synchronous serial interface, use the **dialer string** command. Dialers pass the string you have defined to the external DCE device. ISDN devices call the number specified in the string.

To specify the telephone number call on a serial interface (asynchronous or synchronous), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer string dial-string[:isdn-subaddress]`	Specifies the number to dial.

Dialers pass the string (telephone number) to the external DCE device, which dials the number; ISDN devices themselves call the specified number.

Specifying Chat Scripts and Dial Strings for Asynchronous Serial Interfaces

The modem chat script becomes the default chat script for an interface, which means it becomes the default chat script for the **dialer string** and **dialer map** commands presented in this section.

To place a call to a single site on an asynchronous line for which either a modem dialing script has not been assigned or a system script login must be specified, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer map protocol next-hop-address [modem-script modem-regexp] [system-script system-regexp] dial-string [:isdn-subaddress]`	Specifies chat scripts and a dial string.

Refer to the sections "How To Configure Chat Scripts" and "Dialer Mapping Example" in the chapter "Creating and Using Modem Chat Scripts" for more information about configuring chat scripts.

Configuring the Interface to Receive Calls

If you enable DDR on an interface by using the **dialer in-band** command, the interface can receive calls. No additional configuration steps are required simply to receive calls. Parity is not required for receiving calls only. An interface configured with the **dialer in-band** command can terminate calls when the line is idle for some configurable time.

You cannot set up an ISDN interface only to receive calls from a single site, but you can set it up to receive and place calls to a single site.

To receive calls from an interface that is using DTR dialing, an interface can be configured for in-band dialing or not configured for anything but encapsulation, depending on the desired behavior. If you expect the receiving interface to terminate a call when no traffic is received for some time, you must configure in-band dialing (along with access lists and a dummy dialer string). If the receiving interface is purely passive, no additional configuration is necessary.

Authentication is not required when traffic comes from only one site. However, you can configure authentication for security. See the "Defining the Traffic to Be Authenticated" section. If you want to receive calls *only*, do not provide a dial string in the **dialer map** command shown in that section.

Configuring the Interface to Place and Receive Calls

If you enable DDR on an interface by using the **dialer in-band** command, the interface can receive calls. To enable it to place calls to one site, you must define the dialing destination.

To define the dialing destination, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer string` *dial-string*`[:`*isdn-subaddress*`]`	Specifies the number to dial one site.

Note Use the **dialer map** command instead of the **dialer string** command if you want to authenticate calls received. See the section "Defining the Traffic to Be Authenticated" later in this chapter for more information.

When a dialer string is configured but PPP Challenge Handshake Authentication Protocol (CHAP) is not configured on the interface, the Cisco IOS software recognizes each incoming call as coming from the configured dialer string. That is, if your outgoing calls go to only one number and you do not authenticate incoming calls, it is assumed that all incoming calls come from that number. (If you received calls from multiple sites, you would need to authenticate the calls.)

Authentication is not required when traffic comes from only one site. However, you can configure authentication for an extra measure of security. See the following section, "Defining the Traffic to Be Authenticated," for more information. If you want to receive and place calls, use the **dialer map** command.

Defining the Traffic to Be Authenticated

Authentication can be done through CHAP or Password Authentication Protocol (PAP). In addition, the interface must be configured to map the protocol address of the host to the name to use for authenticating the remote host.

To enable CHAP or PAP on an interface and authenticate sites that are calling in, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Configures an interface for PPP encapsulation.

Step 2	`Router(config-if)# ppp authentication chap [if-needed]`
	or
	`Router(config-if)# ppp authentication pap [if-needed]`
		Enables CHAP.
		
		Enables PAP.

Step 3	`Router(config-if)# dialer map` *protocol next-hop-address* **name** *hostname* [**modem-script** *modem-regexp*] [**system-script** *system-regexp*] [*dial-string*[**:***isdn-subaddress*]]	Maps the protocol address to a host name.

If the dial string is not provided in the chat script, the interface will be able to receive calls from the host but will not be able to place calls to the host.

Configuring Access Control for Outgoing Calls

Protocol access lists and dialer access lists are central to the operation of DDR. In general, access lists are used as the screening criteria for determining when to initiate DDR calls. All packets are tested against the dialer access list. Packets that match a permit entry are deemed *interesting*. Packets that do not match a permit entry or that do match a deny entry are deemed *uninteresting*. When a packet is found to be interesting, either the dialer idle timer is reset (if the line is active) or a connection is attempted (if the line is available but not active). If a tested packet is deemed *uninteresting*, it will be forwarded if it is intended for a destination known to be on a specific interface and the link is active. However, such a packet will not initiate a DDR call and will not reset the idle timer.

Configuring Access Control for Bridging

You can control access by defining any transparent bridge packet as *interesting*, or you can use the finer granularity of controlling access by Ethernet type codes. To control access for DDR bridging, perform *one* of the following tasks in global configuration mode:

- Controlling Bridging Access by Ethernet Type Codes (As required)
- Permitting All Bridge Packets to Trigger Calls (As required)
- Assigning the Interface to a Bridge Group (As required)

Note Spanning-tree bridge protocol data units (BPDUs) are always treated as *uninteresting*.

Controlling Bridging Access by Ethernet Type Codes

To control access by Ethernet type codes, use the following command in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {permit | deny} type-code [mask]`	Identifies *interesting* packets by Ethernet type codes (access list numbers must be in the range 200 to 299).

To enable packets with a specified Ethernet type code to trigger outgoing calls, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer-list dialer-group protocol bridge list access-list-number`	Defines a dialer list for the specified access list.

For a table of some common Ethernet types codes, see the "Ethernet Types Codes" appendix in the *Cisco IOS Bridging and IBM Networking Command Reference.*

Permitting All Bridge Packets to Trigger Calls

To identify all transparent bridge packets as interesting, use the following command in interface configuration mode when you are configuring DDR:

Command	Purpose
`Router(config-if)# dialer-list dialer-group protocol bridge permit`	Defines a dialer list that treats all transparent bridge packets as *interesting*.

Assigning the Interface to a Bridge Group

Packets are bridged only among interfaces that belong to the same bridge group. To assign an interface to a bridge group, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# bridge-group bridge-group`	Assigns the specified interface to a bridge group.

Configuring Access Control for Routing

Before you perform the tasks outlined in this section, configure access lists for the protocols you intend to route over DDR as described briefly in the chapter "Preparing to Configure DDR" in this publication, and as described in greater detail in the appropriate network protocol configuration guide (for example, the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*).

An interface can be associated only with a single dialer access group; multiple dialer access group assignments are not allowed. To specify the dialer access group to which you want to assign an access list, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer-group group-number`	Specifies the number of the dialer access group to which the specific interface belongs.

Customizing the Interface Settings

To customize DDR in your network, perform the tasks in the following sections as needed:

- Configuring Timers on the DDR Interface (As required)
- Setting Dialer Interface Priority (As required)
- Configuring a Dialer Hold Queue (As required)
- Configuring Bandwidth on Demand (As required)
- Disabling and Reenabling DDR Fast Switching (As required)
- Configuring Dialer Redial Options (As required)

Configuring Timers on the DDR Interface

To set the timers, perform the tasks in the following sections as needed:

- Setting Line-Idle Time (As required)
- Setting Idle Time for Busy Interfaces (As required)
- Setting Line-Down Time (As required)
- Setting Carrier-Wait Time (As required)

Setting Line-Idle Time

To specify the amount of time for which a line will stay idle before it is disconnected, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer idle-timeout seconds [inbound | either]`	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.

Note The **dialer idle-timeout** interface configuration command specifies the duration of time before an idle connection is disconnected. Previously, both inbound and outbound traffic would reset the dialer idle timer; now you can specify that only inbound traffic will reset the dialer idle timer.

Setting Idle Time for Busy Interfaces

The dialer fast idle timer is activated if there is contention for a line. Contention occurs when a line is in use, a packet for a different next hop address is received, and the busy line is required to send the competing packet.

If the line has been idle for the configured amount of time, the current call is disconnected immediately and the new call is placed. If the line has not yet been idle as long as the fast idle timeout period, the packet is dropped because there is no way to get through to the destination. (After the packet is dropped, the fast idle timer remains active and the current call is disconnected as soon as it has been idle for as long as the fast idle timeout.) If, in the meantime, another packet is sent to the currently connected destination, and it is classified as interesting, the fast-idle timer is restarted.

To specify the amount of time for which a line for which there is contention will stay idle before the line is disconnected and the competing call is placed, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer fast-idle seconds`	Sets idle time for high traffic lines.

This command applies to both inbound and outbound calls.

Setting Line-Down Time

To set the length of time for which the interface stays down before it is available to dial again after a line is disconnected or fails, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer enable-timeout seconds`	Sets the interface downtime.

This command applies to both inbound and outbound calls.

Setting Carrier-Wait Time

To set the length of time for which an interface waits for the telephone service (carrier), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer wait-for-carrier-time seconds`	Sets the length of time for which the interface waits for the carrier to come up when a call is placed.

For asynchronous interfaces, this command sets the total time to wait for a call to connect. This time is set to allow for running the chat script.

Setting Dialer Interface Priority

Interface priority indicates which interface in a dialer rotary group will get used first for outgoing calls. You might give one interface a higher priority if it is attached to a faster, more reliable modem. In this way, the higher-priority interface will be used as often as possible.

To assign priority to an interface in a dialer rotary group, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer priority number`	Sets the interface priority in the dialer rotary group.

The range of values for *number* is 0 through 255. Zero is the default value and lowest priority; 255 is the highest priority. This command applies to outgoing calls only.

Configuring a Dialer Hold Queue

Sometimes packets destined for a remote router are discarded because no connection exists. Establishing a connection using an analog modem can take time, during which packets are discarded. However, configuring a dialer hold queue will allow *interesting* outgoing packets to be queued and sent as soon as the modem connection is established.

A dialer hold queue can be configured on any type of dialer, including in-band synchronous, asynchronous, DTR, and ISDN dialers. Also, *hunt group leaders* can be configured with a dialer hold queue. If a hunt group leader (of a rotary dialing group) is configured with a hold queue, all members of the group will be configured with a dialer hold queue and no hold queue of an individual member can be altered.

To establish a dialer hold queue, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer hold-queue packets`	Creates a dialer hold queue and specifies the number of packets to be held in it.

As many as 100 packets can be held in an outgoing dialer hold queue.

Configuring Bandwidth on Demand

You can configure a dialer rotary group to use additional bandwidth by placing additional calls to a single destination if the load for the interface exceeds a specified weighted value. Parallel communication links are established based on traffic load. The number of parallel links that can be established to one location is not limited.

To set the dialer load threshold for bandwidth on demand, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer load-threshold load`	Configures the dialer rotary group to place additional calls to a single destination, as indicated by interface load.

Once multiple links are established, they are still governed by the load threshold. If the total load on all the links falls below the threshold, an idle link will be torn down.

Disabling and Reenabling DDR Fast Switching

Fast switching is enabled by default on all DDR interfaces. When fast switching is enabled or disabled on an ISDN D channel, it is enabled or disabled on all B channels. When fast switching is enabled or disabled on a dialer interface, it is enabled or disabled on all rotary group members but cannot be enabled or disabled on the serial interfaces individually.

Fast switching can be disabled and re-enabled on a protocol-by-protocol basis. To disable fast switching and re-enable it, use one of the following protocol-specific commands in interface configuration mode:

Command	Purpose
`Router(config-if)# no ip route-cache`	Disables IP fast switching over a DDR interface.
`Router(config-if)# ip route cache`	Reenables IP fast switching over a DDR interface.
`Router(config-if)# no ip route-cache distributed`	Disables distributed IP fast switching over a DDR interface. This feature works in Cisco 7500 routers with a Versatile Interface Processor (VIP) card.
`Router(config-if)# ip route-cache distributed`	Enables distributed IP fast switching over a DDR interface. This feature works in Cisco 7500 routers with a VIP card.
`Router(config-if)# no ipx route-cache`	Disables IPX fast switching over a DDR interface.
`Router(config-if)# ipx route-cache`	Reenables IPX fast switching over a DDR interface.

Configuring Dialer Redial Options

By default, the Cisco IOS software generates a single dial attempt for each interesting packet. Dialer redial allows the dialer to be configured to make a maximum number of redial attempts if the first dial-out attempt fails, wait a specific interval between redial attempts, and disable the interface for a specified duration if all redial attempts fail. New dialout attempts will not be initiated if a redial is pending to the same destination.

To configure redial options, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer`	Enters interface configuration mode.
Step 2	`Router(config-if)# dialer redial interval` *time* `attempts` *number* `re-enable` *disable-time*	Configures redial options on the router.

Sending Traffic over Frame Relay, X.25, or LAPB Networks

An interface configured for DDR can send traffic over networks that require Link Access Procedure, Balanced (LAPB), X.25, or Frame Relay encapsulation.

Before Cisco IOS software Release 12.0(6)T, encapsulation techniques such as Frame Relay, HDLC, LAPB-TA, and X.25 could support only one ISDN B-channel connection over the entire link. HDLC and PPP could support multiple B channels, but the entire ISDN link needed to use the same encapsulation. The Dynamic Multiple Encapsulations feature allows incoming calls over ISDN to be assigned encapsulation type based on calling line identification (CLID) or DNIS. With the Dynamic Multiple Encapsulations feature, once CLID binding is completed, the topmost interface is always used for all

configuration and data structures. The ISDN B channel becomes a forwarding device, and the configuration on the D channel is ignored, thereby allowing the different encapsulation types and per-user configurations.

To configure an interface for those networks, perform the tasks in the following sections:

- Configuring the Interface for Sending Traffic over a Frame Relay Network (As required)
- Configuring the Interface for Sending Traffic over an X.25 Network (As required)
- Configuring the Interface for Sending Traffic over a LAPB Network (As required)

Configuring the Interface for Sending Traffic over a Frame Relay Network

Access to Frame Relay networks is now available through dialup connections as well as leased lines. Dialup connectivity allows Frame Relay networks to be extended to sites that do not generate enough traffic to justify leased lines, and also allows a Frame Relay network to back up another network or point-to-point line.

DDR over Frame Relay is supported for synchronous serial and ISDN interfaces and for rotary groups, and is available for in-band, DTR, and ISDN dialers.

Frame Relay supports multiple permanent virtual circuit (PVC) connections over the same serial interface or ISDN B channel, but only one *physical* interface can be used (dialed, connected, and active) in a rotary group or with ISDN.

The Dynamic Multiple Encapsulations feature supports the following Frame Relay features:

- Frame Relay RTP Header Compression (RFC 1889)
- Frame Relay TCP/IP Header Compression
- Legacy DDR over Frame Relay
- Frame Relay Interface/Subinterface Backup

Dynamic multiple encapsulations support at least four Frame Relay PVCs on either dialer interfaces or dialer subinterfaces.

Note Frame Relay encapsulations in the Dynamic Multiple Encapsulations feature do not support IETF or Cisco Encapsulation for IBM Systems Network Architecture (SNA). Frame Relay for SNA support is not applicable.

Configuration Restrictions

The following restrictions apply to DDR used over Frame Relay:

- Frame Relay is not available for asynchronous dialers.
- The Frame Relay Dynamic Multiple Encapsulations feature does *not* provide bidirectional support.
- With the Dynamic Multiple Encapsulations feature, there is no process switching for Frame Relay packets; these packets are always fast switched.
- Like HDLC, LAPB, and X.25, Frame Relay does not provide authentication. However, ISDN dialers can offer some authentication through the caller ID feature.
- Only one ISDN B channel can be dialed at any one time. When configuring a rotary group, you can use only one serial interface.

Frame Relay subinterfaces work the same on dialup connections as they do on leased lines.

Configuration Overview

No new commands are required to support DDR over Frame Relay. In general, you configure Frame Relay and configure DDR. In general, complete the following tasks to configure an interface for DDR over Frame Relay:

- Specify the interface.
- Specify the protocol identifiers for the interface.

  For example, enter the IP address and mask, the IPX network number, and the AppleTalk cable range and zone.
- Configure Frame Relay.

  As a minimum, you must enable Frame Relay encapsulation and decide whether you need to do static or dynamic address mapping. If you decide to do dynamic mapping, you need not enter a command because Inverse Address Resolution Protocol is enabled by default. If you decide to do static mapping, you must enter Frame Relay mapping commands.

  You can then configure various options as needed for your Frame Relay network topology.
- Configure DDR.

  At a minimum, you must decide and configure the interface for outgoing calls only, incoming calls only, or both outgoing and incoming calls.

  You can also configure DDR for your routed protocols (as specified in the section "Preparations for Routing or Bridging over DDR" in the chapter "Preparing to Configure DDR" in this publication) and for snapshot routing (as specified in the chapter "Configuring Snapshot Routing" later in this publication). You can also customize DDR interfaces on your router or access server (as described in the section "Customizing the Interface Settings" in this chapter).

For examples of configuring various interfaces for DDR over Frame Relay, see the section "Frame Relay Support Example" later in this chapter.

Configuring the Interface for Sending Traffic over an X.25 Network

X.25 interfaces can now be configured to support DDR. Synchronous serial and ISDN interfaces on Cisco routers and access servers can be configured for X.25 addresses, X.25 encapsulation, and mapping of protocol addresses to the X.25 address of a remote host. In-band, DTR, and ISDN dialers can be configured to support X.25 encapsulation, but rotary groups cannot.

Remember that for ISDN interfaces, once CLID binding is completed, the topmost interface is always used for all configuration and data structures. The ISDN B channel becomes a forwarding device, and the configuration on the D channel is ignored, thereby allowing the different encapsulation types and per-user configurations. For X.25 encapsulations, the configurations reside on the dialer profile. The Dynamic Multiple Encapsulations feature provides support for packet assembler/disassembler (PAD) traffic and X.25 encapsulated and switched packets.

To configure an interface to support X.25 and DDR, use the following X.25-specific commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation x25 [dte | dce] [ietf]`	Configures the interface to use X.25 encapsulation.

	Command	Purpose
Step 2	`Router(config-if)# x25 address x.121-address`	Assigns an X.25 address to the interface.
Step 3	`Router(config-if)# x25 map protocol address [protocol2 address2 [...[protocol9 address9]]] x.121-address [option]`	Sets up the LAN protocols-to-remote host address mapping.

The order of DDR and X.25 configuration tasks is not critical; you can configure DDR before or after X.25, and you can even mix the DDR and X.25 commands.

For an example of configuring an interface for X.25 encapsulation and then completing the DDR configuration, see the section "X.25 Support Example" later in this chapter.

Configuring the Interface for Sending Traffic over a LAPB Network

DDR over serial lines now supports LAPB encapsulation, in addition to the previously supported PPP, HDLC, and X.25 encapsulations.

LAPB encapsulation is supported on synchronous serial, ISDN, and dialer rotary group interfaces, but not on asynchronous dialers.

Because the default encapsulation is HDLC, you must explicitly configure LAPB encapsulation. To configure an interface to support LAPB encapsulation and DDR, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation lapb [dte | dce] [multi | protocol]`	Specifies LAPB encapsulation.

For more information about the serial connections on which LAPB encapsulation is appropriate, refer to the **encapsulation lapb** command in the chapter "X.25 and LAPB Commands" in the *Cisco IOS Wide-Area Networking Command Reference.*

For an example of configuring an interface for DDR over LAPB, see the section "LAPB Support Example" later in this chapter.

Monitoring DDR Connections

To monitor DDR connections, use any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show dialer [interface type number]`	Displays general diagnostics about the DDR interface.
`Router# show dialer map`	Displays current dialer maps, next-hop protocol addresses, user names, and the interfaces on which they are configured.
`Router# show interfaces bri 0`	Displays information about the ISDN interface.
`Router# show ipx interface [type number]`	Displays status about the IPX interface.
`Router# show ipx traffic`	Displays information about the IPX packets sent by the router or access server, including watchdog counters.

Command	Purpose
`Router# show appletalk traffic`	Displays information about the AppleTalk packets sent by the router or access server.
`Router# show vines traffic`	Displays information about the Banyan VINES packets sent by the router or access server.
`Router# show decnet traffic`	Displays information about the DECnet packets sent by the router or access server.
`Router# show xns traffic`	Displays information about the XNS packets sent by the router or access server.
`Router# clear dialer`	Clears the values of the general diagnostic statistics.

Configuration Examples for Legacy DDR Spoke

The following section provides various DDR configurations examples:

- Legacy Dial-on-Demand Routing Example
- Transparent Bridging over DDR Examples
- DDR Configuration in an IP Environment Example
- Two-Way DDR for Novell IPX Example
- AppleTalk Configuration Example
- DECnet Configuration Example
- ISO CLNS Configuration Example
- XNS Configuration Example
- Single Site Dialing Example
- DTR Dialing Example
- Hub-and-Spoke DDR for Asynchronous Interfaces and Authentication Example
- Two-Way Reciprocal Client/Server DDR Without Authentication Example
- Frame Relay Support Example
- X.25 Support Example
- LAPB Support Example

Legacy Dial-on-Demand Routing Example

The following example shows a Cisco 2600 series router that has enabled the **dialer idle-timeout** command with the **inbound** keyword. This command allows only inbound traffic that conforms to the dialer list to establish a connection and reset the dialer idle timer.

```
interface BRI0/0
 ip address 10.1.1.1 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 dialer idle-timeout 120 inbound
 dialer map ip 10.1.1.2 name 2611-7 0201
 dialer-group 1
```

```
  isdn switch-type basic-5ess
  no cdp enable
  ppp authentication chap
!
 ip classless
 ip route 10.2.1.1 255.255.255.255 10.1.1.2
!
access-list 101 permit icmp any any
 access-list 101 deny   ip any any
 dialer-list 1 protocol ip list 101
 tftp-server flash c2600-i-mz.jtong-CSCdm88145-120
```

Transparent Bridging over DDR Examples

The following two examples differ only in the packets that cause calls to be placed. The first example specifies by protocol (any bridge packet is permitted to cause a call to be made); the second example allows a finer granularity by specifying the Ethernet type codes of bridge packets.

The first example configures the serial 1 interface for DDR bridging. Any bridge packet is permitted to cause a call to be placed.

```
no ip routing
!
interface Serial1
 no ip address
 encapsulation ppp
 dialer in-band
 dialer enable-timeout 3
 dialer map bridge name urk broadcast 8985
 dialer hold-queue 10
 dialer-group 1
 ppp authentication chap
 bridge-group 1
 pulse-time 1
!
dialer-list 1 protocol bridge permit
bridge 1 protocol ieee
bridge 1 hello 10
```

The second example also configures the serial 1 interface for DDR bridging. However, this example includes an **access-list** command that specifies the Ethernet type codes that can cause calls to be placed and a **dialer list protocol list** command that refers to the specified access list.

```
no ip routing
!
interface Serial1
 no ip address
 encapsulation ppp
 dialer in-band
 dialer enable-timeout 3
 dialer map bridge name urk broadcast 8985
 dialer hold-queue 10
 dialer-group 1
 ppp authentication chap
 bridge-group 1
 pulse-time 1
!
access-list 200 permit 0x0800 0xFFF8
!
dialer-list 1 protocol bridge list 200
bridge 1 protocol ieee
bridge 1 hello 10
```

DDR Configuration in an IP Environment Example

The following example illustrates how to use DDR on an synchronous interface in an IP environment. You could use the same configuration on an asynchronous serial interface by changing *interface serial 1* to specify an asynchronous interface (for example, *interface async 0*).

```
interface serial 1
ip address 172.18.126.1 255.255.255.0
dialer in-band
! The next command sets the dialer idle time-out to 10 minutes.
dialer idle-timeout 600
! The next command inserts the phone number.
dialer string 5551234
! The next command gives the modem enough time to recognize that
! DTR has dropped so the modem disconnects the call.
pulse-time 1
! The next command adds this interface to the dialer access group defined with
! the dialer-list command.
dialer-group 1
!
! The first access list statement, below, specifies that IGRP updates are not
! interesting packets. The second access-list statement specifies that all
! other IP traffic such as Ping, Telnet, or any other IP packet are interesting
! packets. The dialer-list command then creates dialer access group 1 and states
! that access list 101 is to be used to classify packets as interesting or
! uninteresting. The ip route commands specify that there is a route to network
! 172.18.29.0 and to network 172.18.1.0 via 131.108.126.2. This means that several
! destination networks are available through a router that is dialed from interface
! async 1.
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
ip route 172.18.29.0 172.18.126.2
ip route 172.18.1.0 172.18.126.2
ip local pool dialin 10.102.126.2 10.102.126.254
```

With many modems, the **pulse-time** command must be used so that DTR is dropped for enough time to allow the modem to disconnect.

The **redistribute static** command can be used to advertise static route information for DDR applications. Refer to the **redistribute static ip** command, described in the chapter "IP Routing Commands" of the *Cisco IOS IP Command Reference*. Without this command, static routes to the hosts or network that the router can access with DDR will not be advertised to other routers with which the router is communicating. This behavior can block communication because some routes will not be known.

Two-Way DDR for Novell IPX Example

You can set DDR for Novell IPX so that both the client and server have dial-in access to each other. This configuration is demonstrated in the following two subsections.

Remote Configuration Example

The following example is performed on the remote side of the connection:

```
username local password secret
ipx routing
!
```

```
interface ethernet 0
 ipx network 40
!
interface async
 ip unnumbered e0
 encapsulation ppp
 async mode dedicated
 async dynamic routing
 ipx network 45
 ipx watchdog-spoof
 dialer in-band
 dialer map ipx 45.0000.0cff.d016 broadcast name local 1212
 dialer-group 1
 ppp authentication chap
!
access-list 901 deny 0 FFFFFFFF 452
access-list 901 deny 0 FFFFFFFF 453
access-list 901 deny 0 FFFFFFFF 457
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 457
access-list 901 permit 0
ipx route 41 45.0000.0cff.d016
ipx route 50 45.0000.0cff.d016
ipx sap 4 SERVER 50.0000.0000.0001 451 2
chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
dialer-list 1 list 901
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Local Configuration Example

The following example is performed on the local side of the connection:

```
username remote password secret
ipx routing
!
interface ethernet 0
 ipx network 41
!
interface async
 ip unnumbered e0
 encapsulation ppp
 async mode dedicated
 async dynamic routing
 ipx network 45
 ipx watchdog-spoof
 dialer in-band
 dialer map ipx 45.0000.0cff.d016 broadcast name remote 8888
 dialer-group 1
 ppp authentication chap
!
access-list 901 deny 0 FFFFFFFF 452
access-list 901 deny 0 FFFFFFFF 453
access-list 901 deny 0 FFFFFFFF 457
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny 0 FFFFFFFF 0 FFFFFFFF 457
```

```
access-list 901 permit 0
ipx route 40 45.0000.0cff.d016
chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
dialer-list 1 list 901
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

AppleTalk Configuration Example

The following example configures DDR for AppleTalk access using an ISDN BRI. Two access lists are defined: one for IP and Interior Gateway Routing Protocol (IGRP) and one for AppleTalk. AppleTalk packets from network 2141 only (except broadcast packets) can initiate calls.

```
interface BRI0
 ip address 172.17.20.107 255.255.255.0
 encapsulation ppp
 appletalk cable-range 2141-2141 2141.65
 appletalk zone SCruz-Eng
 no appletalk send-rtmps
 dialer map ip 172.17.20.106 broadcast 1879
 dialer map appletalk 2141.66 broadcast 1879
 dialer-group 1
!
access-list 101 deny   igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 601 permit cable-range 2141-2141 broadcast-deny
access-list 601 deny other-access
!
dialer-list 1 list 101
dialer-list 1 list 601
```

DECnet Configuration Example

The following example configures DDR for DECnet:

```
decnet routing 10.19
username RouterB password 7 030752180531
interface serial 0
 no ip address
 decnet cost 10
 encapsulation ppp
 dialer in-band
 dialer map decnet 10.151 name RouterB broadcast 4155551212
 dialer-group 1
 ppp authentication chap
 pulse-time 1
access-list 301 permit 10.0 0.1023 0.0 63.1023
dialer-list 1 protocol decnet list 301
```

ISO CLNS Configuration Example

The following example configures a router for International Organization for Standardization Connectionless Network Service (ISO CLNS) DDR with in-band dialing:

```
username RouterB password 7 111C140B0E
clns net 47.0004.0001.0000.0c00.2222.00
clns routing
clns filter-set ddrline permit 47.0004.0001....
!
interface serial 0
 no ip address
 encapsulation ppp
 dialer in-band
 dialer map clns 47.0004.0001.0000.0c00.1111.00 name RouterB broadcast 1212
 dialer-group 1
 ppp authentication chap
 clns enable
 pulse-time 1
!
clns route default serial 0
dialer-list 1 protocol clns list ddrline
```

XNS Configuration Example

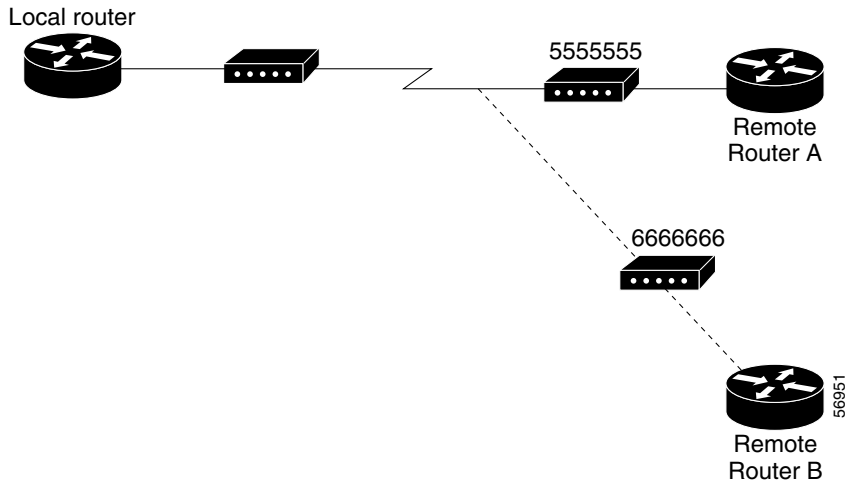
The following example configures DDR for XNS. The access lists deny broadcast traffic to any host on any network, but allow all other traffic.

```
xns routing 0000.0c01.d8dd

username RouterB password 7 111B210A0F

interface serial 0
 no ip address
 encapsulation ppp
 xns network 10
 dialer in-band
 dialer map xns 10.0000.0c01.d877 name RouterB broadcast 4155551212
 dialer-group 1
 ppp authentication chap
 pulse-time 1
!
access-list 400 deny -1 -1.ffff.ffff.ffff 0000.0000.0000
access-list 400 permit -1 10
!
dialer-list 1 protocol xns list 400
```

Single Site Dialing Example

The following example is based on the configuration shown in Figure 49; the router receives a packet with a next hop address of 10.1.1.1.

Figure 49 Sample Dialer String or Dialer Map Configuration



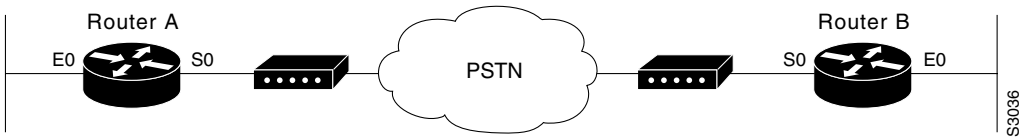
If the single site called by the DDR spoke interface on your router has the phone number 5555555, it will send the packet to that site, assuming that the next hop address 10.1.1.1 indicates the same remote device as phone number 5555555. The **dialer string** command is used to specify the string (telephone number) to be called.

```
interface serial 1
 dialer in-band
 dialer string 5555555
```

DTR Dialing Example

The following example shows Router A and Router B connected to a Public Switched Telephone Network (PSTN). Router A is configured for DTR dialing. Remote Router B is configured for in-band dialing so it can disconnect an idle call. (See Figure 50.)

Figure 50 DTR Dialing Through a PSTN



Router A

```
interface serial 0
 ip address 172.18.170.19 255.255.255.0
 dialer dtr
 dialer-group 1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 101
```

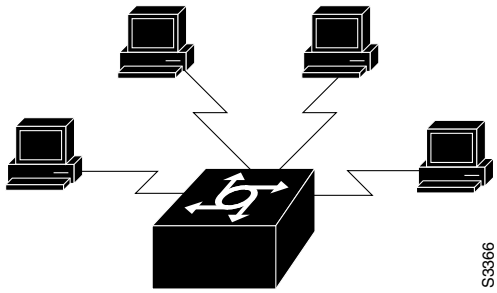
Router B

```
interface serial 0
 ip address 172.18.170.20 255.255.255.0
 dialer in-band
 dialer string 9876543
 pulse-time 1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 list 101
```

Hub-and-Spoke DDR for Asynchronous Interfaces and Authentication Example

The following example sets up DDR to provide service to multiple remote sites. In a hub-and-spoke configuration, you can use a generic configuration script to set up each remote connection. Figure 51 illustrates a typical hub-and-spoke configuration.

Figure 51 Hub-and-Spoke DDR Configuration



Commands in the following sections are used to create this configuration.

Spoke Topology Configuration

The following commands are executed on the spoke side of the connection. (A different "spoke" password must be specified for each remote client.) The configuration provides authentication by identifying a password that must be provided on each end of the connection.

```
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 async mode dedicated
 async default ip address 172.30.45.1
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.1 name hub system-script hub 1234
 dialer map ip 172.30.45.255 name hub system-script hub 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 172.30.45.1
ip default-network 172.30.0.0
chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
```

```
chat-script hub "" "" name: spoke1 word: <spoke1-passwd> PPP
dialer-list 1 protocol ip permit
!
username hub password <spoke1-passwd>
!
router igrp 109
 network 172.30.0.0
 passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Hub Router Configuration

The following commands are executed on the local side of the connection—the hub router. The commands configure the server for communication with three clients and provide authentication by identifying a unique password for each "spoke" in the hub-and-spoke configuration.

```
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode interactive
 async dynamic address
 dialer rotary-group 1
!
interface async 8
 async mode interactive
 async dynamic address
 dialer rotary-group 1
!
interface dialer 1
 ip address 172.30.45.2 255.255.255.0
 no ip split-horizon
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name spoke1 3333
 dialer map ip 172.30.45.2 name spoke2 4444
 dialer map ip 172.30.45.2 name spoke3 5555
 dialer map ip 172.30.45.255 name spoke1 3333
 dialer map ip 172.30.45.255 name spoke2 4444
 dialer map ip 172.30.45.255 name spoke3 5555
 dialer-group 1
!
ip route 172.30.44.0 255.255.255.0 172.30.45.2
ip route 172.30.44.0 255.255.255.0 172.30.45.3
ip route 172.30.44.0 255.255.255.0 172.30.45.4
dialer-list 1 list 101
 access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
 access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
username spoke1 password <spoke1-passwd>
username spoke2 password <spoke2-passwd>
username spoke3 password <spoke3-passwd>
username spoke1 autocommand ppp 172.30.45.2
username spoke2 autocommand ppp 172.30.45.3
username spoke3 autocommand ppp 172.30.45.4
```

```
!
router igrp 109
 network 172.30.0.0
 redistribute static
!
line 7
 login tacacs
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Two-Way Reciprocal Client/Server DDR Without Authentication Example

You can set up two-way reciprocal DDR without authentication in which both the client and server have dial-in access to each other. This configuration is demonstrated in the following two sections.

Remote Configuration

The following commands are executed on the remote side of the connection. This configuration provides authentication by identifying a password that must be provided on each end of the connection.

```
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 ip address 172.30.45.2 255.255.255.0
 async mode dedicated
 async default ip address 172.30.45.1
 encap ppp
 dialer in-band
 dialer string 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 async 7
 ip default-network 172.30.0.0
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
line 7
 no exec
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Local Configuration

The following commands are executed on the local side of the connection. As with the remote side configuration, this configuration provides authentication by identifying a password for each end of the connection.

```
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode dedicated
 async default ip address 172.30.45.2
 encapsulation ppp
```

```
 dialer in-band
 dialer string 1235
 dialer rotary-group 1
!
interface async 8
 async mode dedicated
 async default ip address 172.30.45.2
 dialer rotary-group 1
!
ip route 172.30.44.0 255.255.255.0 async 7
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name remote 4321
 dialer load-threshold 80
!
ip route 172.30.44.0 255.255.255.0 128.150.45.2
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 dialer-list 1 protocol ip permit
!
route igrp 109
network 172.30.0.0
redistribute static
passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Frame Relay Support Example

The examples in this section present various combinations of interfaces, Frame Relay features, and DDR features.

Frame Relay Access with In-Band Dialing (V.25*bis*) and Static Mapping Example

The following example shows how to configure a router for IP over Frame Relay using in-band dialing. A Frame Relay static map is used to associate the next hop protocol address to the data-link connection identifier (DLCI). The dialer string allows dialing to only one destination.

```
interface Serial0
 ip address 10.1.1.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 10.1.1.2 100 broadcast
 dialer in-band
 dialer string 4155551212
 dialer-group 1
!
access-list 101 deny igrp any host 255.255.255.255
access-list 101 permit ip any any
!
dialer-list 1 protocol ip list 101
```

Frame Relay Access with ISDN Dialing and DDR Dynamic Maps Example

The following example shows a BRI interface configured for Frame Relay and for IP, IPX, and AppleTalk routing. No static maps are defined because this setup relies on Frame Relay Local Management Interface (LMI) signaling and Inverse ARP to determine the network addresses-to-DLCI mappings dynamically. (Because Frame Relay Inverse ARP is enabled by default, no command is required.)

```
interface BRI0
 ip address 10.1.1.1 255.255.255.0
 ipx network 100
 appletalk cable-range 100-100 100.1
 appletalk zone ISDN
 no appletalk send-rtmps
 encapsulation frame-relay IETF
 dialer map ip 10.1.1.2 broadcast 4155551212
 dialer map apple 100.2 broadcast 4155551212
 dialer map ipx 100.0000.0c05.33ed broadcast 4085551234
 dialer-group 1
!
access-list 101 deny igrp any host 255.255.255.255
access-list 101 permit ip any any
access-list 901 deny -1 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 457
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 457
access-list 901 permit -1
access-list 601 permit cable-range 100-100 broadcast-deny
access-list 601 deny other-access
!
dialer-list 1 protocol ip list 101
dialer-list 1 protocol novell list 901
dialer-list 1 protocol apple list 601
```

X.25 Support Example

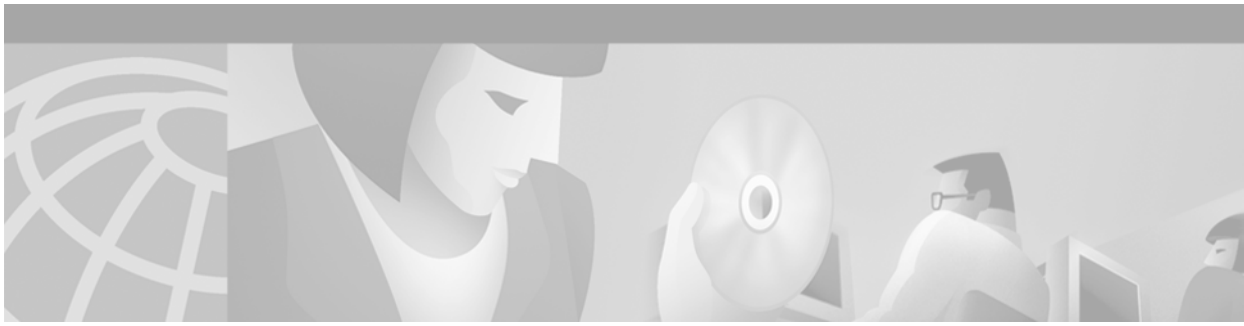
The following example configures a router to support X.25 and DTR dialing:

```
interface serial 0
 ip address 172.18.170.19 255.255.255.0
 encapsulation x25
 x25 address 12345
 x25 map ip 172.18.171.20 67890 broadcast
 dialer dtr
 dialer-group 1
!
 access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
 access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
 dialer-list 1 list 101
```

LAPB Support Example

The following example configures a router for LAPB encapsulation and in-band dialing:

```
interface serial 0
 ip address 172.18.170.19 255.255.255.0
 encapsulation lapb
 dialer in-band
 dialer string 4155551212
 dialer-group 1
!
 access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
 access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
 dialer-list 1 protocol ip list 101
```



Configuring Legacy DDR Hubs

This chapter describes how to configure legacy dial-on-demand routing (DDR) on interfaces functioning as the hub in a hub-and-spoke network topology. It includes the following main sections:

This chapter considers a *hub* interface to be any interface that calls or receives calls from more than one other router and considers a *spoke* interface to be an interface that calls or receives calls from exactly one router.

For configuration tasks for the spoke interfaces in a hub-and-spoke network topology, see the chapter "Configuring Legacy DDR Spokes" in this publication.

For information about the dialer profiles implementation of DDR, see the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles" in this publication.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the DDR commands in this chapter, see the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

DDR Issues

A DDR configuration applies to a specified router interface but serves to meet the communication needs of the network. The router configured for DDR has a function to serve in preserving communications and ensuring that routes are known to other routers at both ends of the dial link. Thus, these issues are important:

- Types and number of router interfaces to be configured for DDR.
- Function of each specific interface—to place calls, receive calls, or both—and the number of sites connecting to the interface.

- Identity and characteristics of the router at the other end of each connection—phone number, host name, next hop network protocol addresses, type of signaling used or required, ability to place or receive calls, other requirements.
- Types of packets that will be allowed to trigger outgoing calls—if the interface places calls.
- End of the connection that will control the communication: initiating calls and terminating calls when the line is idle.
- Method for authenticating other routers—if the interface receives calls from multiple sites.
- Passing routing information across the dial link.

DDR Hubs Configuration Task Flow

Before you configure DDR, make sure you have completed the preparations for bridging or routing as described in the chapter "Preparing to Configure DDR" in this publication. That chapter provides information about the minimal requirements. For detailed information about bridging, routing, and wide-area networking configurations, see the appropriate chapters in other volumes of this documentation set.

When you configure DDR on a hub interface in a hub-and-spoke topology, you perform the following general steps:

Step 1 Specify the interface that will place calls to or receive calls from multiple sites. (See the chapter "Configuring Legacy DDR Spokes" in this publication for information about configuring an interface to place calls to or receive calls from one site only.)

Step 2 Enable DDR on the interface. This step is not required for some interfaces; for example, ISDN interfaces and passive interfaces that receive only from data terminal ready (DTR)-dialing interfaces.

Step 3 Configure the interface to receive calls only, if applicable. Receiving calls from multiple sites requires each inbound call to be authenticated.

Step 4 Configure the interface to place calls only, if applicable.

Step 5 Configure the interface to place and receive calls, if applicable.

Step 6 If the interface will place calls, specify access control for the following:

- Transparent bridging—Assign the interface to a bridge group, and define dialer lists associated with the bridging access lists. The interface switches between members of the same bridge group, and dialer lists specify which packets can trigger calls.

  or

- Routed protocols—Define dialer lists associated with the protocol access lists to specify which packets can trigger calls.

Step 7 Customize the interface settings (timers, interface priority, hold queues, bandwidth on demand, and disabling fast switching) as needed.

When you have configured the interface and it is operational, you can monitor its performance and its connections as described in the "Monitoring DDR Connections" section later in this chapter.

You can also enhance DDR by configuring Multilink PPP and configuring PPP callback. The PPP configuration tasks are described in the chapter "Configuring Media-Independent PPP and Multilink PPP" in this publication.

See the section "Configuration Examples for Legacy DDR Hub" at the end of this chapter for examples of how to configure DDR on your network.

How to Configure DDR

To configure DDR on an interface, perform the tasks in the following sections. The first five bulleted items are required. The remaining tasks are not absolutely required, but might be necessary in your networking environment.

- Specifying the Interface (Required)
- Enabling DDR on the Interface (Required)
- Configuring the Interface to Place Calls Only (Required)
  or
  Configuring the Interface to Receive Calls Only (Required)
  or
  Configuring the Interface to Place and Receive Calls (Required)
- Configuring Access Control for Outgoing Calls (As required)
- Customizing the Interface Settings (As required)
- Sending Traffic over Frame Relay, X.25, or LAPB Networks (As required)

See the section "Monitoring DDR Connections" later in this chapter for commands and other information about monitoring DDR connections. See the section "Configuration Examples for Legacy DDR Hub" at the end of this chapter for ideas about how to implement DDR in your network.

Specifying the Interface

You can configure any asynchronous, synchronous serial, ISDN, or dialer interface for legacy DDR.

Note When you specify an interface, make sure to use the interface numbering scheme supported on the network interface module or other port hardware on the router. On the Cisco 7200 series router, for example, you specify an interface by indicating its type, slot number, and port number.

To specify an interface to configure for DDR, use one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# interface async number`
`Router(config)# interface serial number`
`Router(config)# interface bri number`
or
	Specifies an interface to configure for DDR.

`Router(config)# interface serial slot/port:23` or	Specifies an ISDN PRI D channel (T1).
`Router(config)# interface serial slot/port:15` or	Specifies an ISDN PRI D channel (E1).
`Router(config)# interface dialer number`	Specifies a logical interface to function as a dialer rotary group leader.

Dialer interfaces are logical or virtual entities, but they use physical interfaces to place or receive calls.

Enabling DDR on the Interface

This task is required for asynchronous serial, synchronous serial, and logical dialer interfaces.

This task is not required for ISDN interfaces (BRI interfaces and ISDN PRI D channels) and for *purely passive* interfaces that will receive calls only from interfaces that use DTR dialing.

Enabling DDR on an interface usually requires you to specify the type of dialer to be used. This task is not required for ISDN interfaces because the software automatically configures ISDN interfaces to be dialer type ISDN.

To enable DDR on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer in-band [no-parity | odd-parity]`	Enables DDR on an asynchronous interface or a synchronous serial interface using V.25*bis* modems.

You can optionally specify parity if the modem on this interface uses the V.25*bis* command set. The 1984 version of the V.25*bis* specification states that characters must have odd parity. However, the default for the **dialer in-band** command is no parity.

Configuring the Interface to Place Calls Only

To configure an interface to place calls to multiple destinations, perform the following tasks. The first task is required for all interface types. The second task is required only if you specified a dialer interface.

- Defining the Dialing Destination (Required)
- Specifying a Physical Interface to Use and Assigning It to a Dialer Rotary Group (As required)

Defining the Dialing Destination

For calling multiple sites, an interface or dialer rotary group must be configured to map each next hop protocol address to the dial string (some form of a telephone number) used to reach it.

To define each dialing destination, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer map` *protocol next-hop-address dial-string*`[:`*isdn-subaddress*`]`	Defines a dialing destination for a synchronous serial interface or a dialer interface.
`Router(config-if)# dialer map` *protocol next-hop-address* `[spc] [speed 56 | 64] [broadcast]` `[`*dial-string*`[:`*isdn-subaddress*`]]`	Defines a dialing destination for an ISDN interface (including an ISDN PRI D channel).
`Router(config-if)# dialer map` *protocol next-hop-address* `[modem-script` *modem-regexp*`]` `[system-script` *system-regexp*`]` *dial-string*`[:`*isdn-subaddress*`]`	Defines a dialing destination for an asynchronous interface. If a modem dialing chat script has not been assigned to the line or a system login chat script must be specified, defines both a dialing destination and the chat scripts to use.

Repeat this task as many times as needed to ensure that all dialing destinations are reachable via some next hop address and dialed number.

If you intend to send traffic over other types of networks, see one of the following sections later in this chapter: "Configuring the Interface for Sending Traffic over a Frame Relay Network," "Configuring the Interface for Sending Traffic over an X.25 Network," or "Configuring the Interface for Sending Traffic over a LAPB Network."

Specifying a Physical Interface to Use and Assigning It to a Dialer Rotary Group

This section applies only if you specified a dialer interface to configure for DDR.

To assign a physical interface to a dialer rotary group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial` *number* or `Router(config)# interface async` *number*	Specifies a physical interface to use and begins interface configuration mode.
Step 2	`Router(config-if)# dialer rotary-group` *number*	Assigns the specified physical interface to a dialer rotary group.

Repeat these two steps for each physical interface to be used by the dialer interface.

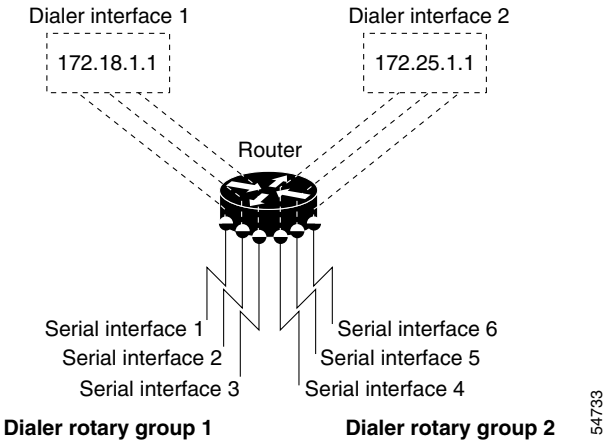
An ISDN BRI is a rotary group of B channels. An ISDN interface can be part of a rotary group comprising other interfaces (synchronous, asynchronous, ISDN BRI, or ISDN PRI). However, Cisco supports at most one level of recursion; that is, a rotary of rotaries is acceptable, but a rotary of rotaries of rotaries is not supported.

Interfaces in a dialer rotary group do not have individual addresses; when the interface is being used for dialing, it inherits the parameters configured for the dialer interface. However, if the individual interface is configured with an address and it is subsequently used to establish a connection from the user EXEC level, the individual interface address again applies.

Note When you look at your configuration file, commands will not appear in the order in which you entered them. You will also see interface configuration commands that you did not enter, because each interface assigned to a dialer rotary group inherits the parameters of the dialer interface in the dialer rotary group.

Figure 52 illustrates how dialer interfaces work. In this configuration, serial interfaces 1, 2, and 3 are assigned to dialer rotary group 1 and thereby take on the parameters configured for dialer interface 1. When it is used for dialing, the IP address of serial interface 2 is the same as the address of the dialer interface, 172.18.1.1.

Figure 52 Sample Dialer Interface Configuration



Configuring the Interface to Receive Calls Only

Once DDR is enabled on an asynchronous serial, synchronous serial, and ISDN interface, the interface can receive calls from multiple sites using one line or multiple lines. However, interfaces that receive calls from multiple sites require authentication of the remote sites. In addition, dialer interfaces require at least one physical interface to be specified and added to the dialer rotary group. The tasks in the following sections describe how to configuration authentication:

- Configuring the Interface for TACACS+

  or

- Configuring the Interface for PPP Authentication
- Specifying Physical Interfaces and Assigning Them to the Dialer Rotary Group

Configuring the Interface for TACACS+

To configure TACACS as an alternative to host authentication, use one of the following commands in interface configuration mode:

Command	Purpose

`Router(config-if)# ppp use-tacacs [single-line]`
or
`Router(config-if)# aaa authentication ppp`
	Configures TACACS.

Use the **ppp use-tacacs** command with TACACS and extended TACACS. Use the **aaa authentication ppp** command with authentication, authorization, and accounting (AAA)/TACACS+.

Configuring the Interface for PPP Authentication

This section specifies the minimum required configuration for PPP Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP) authentication. For more detailed information, see the chapter "Configuring Media-Independent PPP and Multilink PPP" in this publication.

To use CHAP or PAP authentication, perform the following steps beginning in interface configuration mode:

Note After you have enabled one of these protocols, the local router or access server requires authentication of the remote devices that are calling. If the remote device does not support the enabled authentication protocol, no traffic will be passed to that device.

1. For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.
2. Map the protocol address to the name of the host calling in.

To enable PPP encapsulation, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables PPP on an interface.

Step 2	`Router(config-if)# ppp authentication chap [if-needed]`
	or
	`Router(config-if)# ppp authentication pap`
		Enables CHAP on an interface.
		Enables PAP on an interface.

Step 3	`Router(config-if)# dialer map` *protocol next-hop-address* **name** *hostname*	For any host calling in to the local router or access server, maps its host name (case-sensitive) to the next hop address used to reach it. Repeat this step for each host calling in to this interface.

	Command	Purpose
Step 4	`Router(config-if)# exit`	Returns to global configuration mode.
Step 5	`Router(config)# username name [user-maxlinks link-number] password secret`	Specifies the password to be used in CHAP caller identification. Optionally, you can specify the maximum number of connections a user can establish. To use the **user-maxlinks** keyword, you must also use the **aaa authorization network default local** command, and PPP encapsulation and name authentication on all the interfaces the user will be accessing. Repeat this step to add a **username** entry for each remote system from which the local router or access server requires authentication.

Specifying Physical Interfaces and Assigning Them to the Dialer Rotary Group

To assign a physical interface to a dialer rotary group, use the following commands beginning in global configuration mode:

	Command	Purpose

Step 1	`Router(config)# interface serial number`
	or
	`Router(config)# interface async number`
		Specifies a physical interface to use and begins interface configuration mode.

Step 2	`Router(config-if)# dialer rotary-group number`	Assigns the specified physical interface to a dialer rotary group.

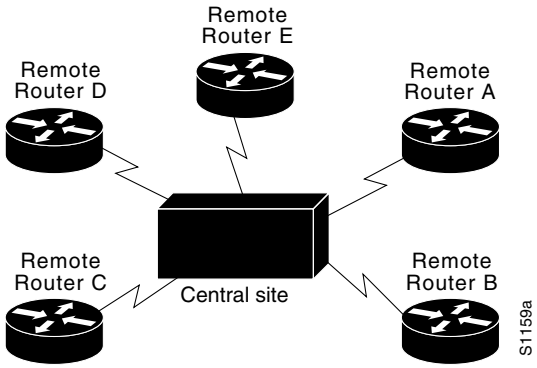
Repeat these two steps for each physical interface to be used by the dialer interface.

Configuring the Interface to Place and Receive Calls

You can configure an physical interface or dialer interface to both place and receive calls. For placing calls, the interface must be configured to map each next hop address to the telephone number to dial. For receiving calls from multiple sites, the interface must be configured to authenticate callers.

Figure 53 shows a configuration in which the central site is calling and receiving calls from multiple sites. In this configuration, multiple sites are calling in to a central site, and the central site might be calling one or more of the remote sites.

Figure 53 Hub-and-Spoke Configuration Using DDR



To configure a single line, multiple lines, or a dialer interface to place calls to and receive calls from multiple sites, perform the tasks in the following section:

- Defining One or More Dialing Destinations
- Defining the Traffic to Be Authenticated

If you intend to send traffic over other types of networks, see one of the following sections later in this chapter: "Configuring the Interface for Sending Traffic over a Frame Relay Network," "Configuring the Interface for Sending Traffic over an X.25 Network," or "Configuring the Interface for Sending Traffic over a LAPB Network."

Defining One or More Dialing Destinations

For calling multiple sites, an interface or dialer rotary group must be configured to map each next hop protocol address to the dial string (some form of a telephone number) used to reach it.

To define each dialing destination, use one of the following commands in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer string dial-string[:isdn-subaddress]`	Defines only one dialing destination (used to configure one phone number on multiple lines only).
`Router(config-if)# dialer map protocol next-hop-address dial-string[:isdn-subaddress]`	Defines one of several dialing destinations for a synchronous serial interface or a dialer interface.
`Router(config-if)# dialer map protocol next-hop-address [spc] [speed 56 | 64][broadcast] [dial-string[:isdn-subaddress]]`	Defines one of several dialing destinations for an ISDN interface (including an ISDN PRI D channel).
`Router(config-if)# dialer map protocol next-hop-address [modem-script modem-regexp] [system-script system-regexp] dial-string[:isdn-subaddress]`	Defines one of several dialing destinations for an asynchronous interface. If a modem dialing chat script has not been assigned to the line or a system login chat script must be specified, define both a dialing destination and the chat scripts to use.

Repeat this task as many times as needed to ensure that all dialing destinations are reachable via some next hop address and dialed number.

Defining the Traffic to Be Authenticated

Calls from the multiple sites must be authenticated. Authentication can be done through CHAP or PAP. In addition, the interface must be configured to map the protocol address of a host to the name to use for authenticating the remote host.

To enable CHAP or PAP on an interface and authenticate sites that are calling in, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Configures an interface for PPP encapsulation.

Step 2	`Router(config-if)# ppp authentication chap [if-needed]`
	or
	`Router(config-if)# ppp authentication pap [if-needed]`
		Enables CHAP.
		Enables PAP.

Step 3	`Router(config-if)# dialer map` *protocol next-hop-address* `name` *hostname* `[modem-script` *modem-regexp*`] [system-script` *system-regexp*`] [`*dial-string*`[:`*isdn-subaddress*`]]`	Maps the protocol address to a host name.

If the dial string is not used, the interface will be able to receive calls from the host, but will not be able to place calls to the host.

Repeat this task for each site from which the router will receive calls.

Configuring Access Control for Outgoing Calls

Protocol access lists and dialer access lists are central to the operation of DDR. In general, access lists are used as the screening criteria for determining when to initiate DDR calls. All packets are tested against the dialer access list. Packets that match a permit entry are deemed *interesting* or *packets of interest*. Packets that do not match a permit entry or that do match a deny entry are deemed *uninteresting*. When a packet is found to be interesting, either the dialer idle timer is reset (if the line is active) or a connection is attempted (assuming the line is available but not active). If a tested packet is deemed *uninteresting*, it will be forwarded if it is intended for a destination known to be on a specific interface and the link is active. However, such a packet will not initiate a DDR call and will not reset the idle timer.

Configuring Access Control for Bridging

When you completed preparations for bridging over DDR, you entered global access lists to specify the protocol packets to be permitted or denied, and global dialer lists to specify which access list to use and which dialer group will place the outgoing calls.

Now you must tie those global lists to an interface configured for DDR. You do this by assigning selected interfaces to a bridge group. Because packets are bridged only among interfaces that belong to the same bridge group, you need to assign this interface and others to the same bridge group.

To assign an interface to a bridge group, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# bridge-group bridge-group`	Assigns the specified interface to a bridge group.

For examples of bridging over DDR, see the "Transparent Bridging over DDR Examples" section later in this chapter.

Configuring Access Control for Routing

Before you perform the tasks outlined in this section, you should have completed the preparations for routing a protocol over DDR as described briefly in the chapter "Preparing to Configure DDR" in this publication and as described in greater detail in the appropriate network protocols configuration guide (for example, the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*).

An interface can be associated only with a single dialer access group; multiple dialer access group assignments are not allowed. To specify the dialer access group to which you want to assign an access list, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer-group group-number`	Specifies the number of the dialer access group to which the specific interface belongs.

Customizing the Interface Settings

To customize DDR in your network, perform the tasks in the following sections as needed:

- Configuring Timers on the DDR Interface (As required)
- Setting Dialer Interface Priority (As required)
- Configuring a Dialer Hold Queue (As required)
- Configuring Bandwidth on Demand (As required)
- Disabling and Reenabling DDR Fast Switching (As required)
- Configuring Dialer Redial Options (As required)

Configuring Timers on the DDR Interface

To configure DDR interface timers, perform the tasks in the following sections as needed:

- Setting Line-Idle Time (As required)
- Setting Idle Time for Busy Interfaces (As required)
- Setting Line-Down Time (As required)
- Setting Carrier-Wait Time (As required)

Setting Line-Idle Time

To specify the amount of time for which a line will stay idle before it is disconnected, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer idle-timeout seconds`	Sets line-idle time.

Setting Idle Time for Busy Interfaces

The dialer fast idle timer is activated if there is contention for a line. Contention occurs when a line is in use, a packet for a different next hop address is received, and the busy line is required to send the competing packet.

If the line has been idle for the configured amount of time, the current call is disconnected immediately and the new call is placed. If the line has not yet been idle as long as the fast idle timeout period, the packet is dropped because the destination is unreachable. (After the packet is dropped, the fast idle timer remains active and the current call is disconnected as soon as it has been idle for as long as the fast idle timeout). If, in the meantime, another packet is sent to the currently connected destination, and it is classified as interesting, the fast-idle timer is restarted.

To specify the amount of time for which a line for which there is contention will stay idle before the line is disconnected and the competing call is placed, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer fast-idle seconds`	Sets idle time for high traffic lines.

This command applies to both inbound and outbound calls.

Setting Line-Down Time

To set the length of time for which the interface stays down before it is available to dial again after a line is disconnected or fails, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer enable-timeout seconds`	Sets the interface downtime.

This command applies to both inbound and outbound calls.

Setting Carrier-Wait Time

To set the length of time for which an interface waits for the telephone service (carrier), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer wait-for-carrier-time seconds`	Sets the length of for which time the interface waits for the carrier to come up when a call is placed.

For asynchronous interfaces, this command sets the total time to wait for a call to connect. This time is set to allow for running the chat script.

Setting Dialer Interface Priority

You can assign dialer priority to an interface. Priority indicates which interface in a dialer rotary group will get used first. To assign priority to a dialer interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer priority` *number*	Specifies which dialer interfaces will be used first.

For example, you might give one interface in a dialer rotary group higher priority than another if it is attached to a faster, more reliable modem. In this way, the higher-priority interface will be used as often as possible.

The range of values for *number* is 0 through 255. Zero is the default value and lowest priority; 255 is the highest priority. This command applies to outgoing calls only.

Configuring a Dialer Hold Queue

Sometimes packets destined for a remote router are discarded because no connection exists. Establishing a connection using an analog modem can take time, during which packets are discarded. However, configuring a dialer hold queue will allow *interesting* outgoing packets to be queued and sent as soon as the modem connection is established.

A dialer hold queue can be configured on any type of dialer, including in-band synchronous, asynchronous, DTR, and ISDN dialers. Also, *hunt group leaders* can be configured with a dialer hold queue. If a hunt group leader (of a rotary dialing group) is configured with a hold queue, all members of the group will be configured with a dialer hold queue and no hold queue for an individual member can be altered.

To establish a dialer hold queue, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer hold-queue` *packets*	Creates a dialer hold queue and specifies the number of packets to be held in it.

As many as 100 packets can be held in an outgoing dialer hold queue.

Configuring Bandwidth on Demand

You can configure a dialer rotary group to use additional bandwidth by placing additional calls to a single destination if the load for the interface exceeds a specified weighted value. Parallel communication links are established based on traffic load. The number of parallel links that can be established to one location is not limited.

To set the dialer load threshold for bandwidth on demand, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer load-threshold load`	Configures the dialer rotary group to place additional calls to a destination, as indicated by interface load.

Once multiple links are established, they are still governed by the load threshold. If the total load falls below the threshold, an idle link will be torn down.

Disabling and Reenabling DDR Fast Switching

Fast switching is enabled by default on all DDR interfaces. When fast switching is enabled or disabled on an ISDN D channel, it is enabled or disabled on all B channels. When fast switching is enabled or disabled on a dialer interface, it is enabled or disabled on all rotary group members but cannot be enabled or disabled on the serial interfaces individually.

Fast switching can be disabled and re-enabled on a protocol-by-protocol basis. To disable fast switching and re-enable it, use one of the following protocol-specific commands in interface configuration mode:

Command	Purpose
`Router(config-if)# no ip route-cache`	Disables IP fast switching over a DDR interface.
`Router(config-if)# ip route cache`	Reenables IP fast switching over a DDR interface.
`Router(config-if)# no ip route-cache distributed`	Disables distributed IP fast switching over a DDR interface. This feature works in Cisco 7500 routers with a Versatile Interface Processor (VIP) card.
`Router(config-if)# ip route-cache distributed`	Enables distributed IP fast switching over a DDR interface. This feature works in Cisco 7500 routers with a VIP card.
`Router(config-if)# no ipx route-cache`	Disables IPX fast switching over a DDR interface.
`Router(config-if)# ipx route-cache`	Reenables IPX fast switching over a DDR interface.

Configuring Dialer Redial Options

By default, the Cisco IOS software generates a single dial attempt for each interesting packet. Dialer redial allows the dialer to be configured to make a maximum number of redial attempts if the first dial-out attempt fails, wait a specific interval between redial attempts, and disable the interface for a specified duration if all redial attempts fail. New dialout attempts will not be initiated if a redial is pending to the same destination.

To configure redial options, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer`	Enters interface configuration mode.
Step 2	`Router(config-if)# dialer redial interval` *time* `attempts` *number* `re-enable` *disable-time*	Configures redial options on the router.

Sending Traffic over Frame Relay, X.25, or LAPB Networks

An interface configured for DDR can send traffic over networks that require Link Access Procedure, Balanced (LAPB), X.25, or Frame Relay encapsulation.

Before Cisco IOS software Release 12.0(6)T, encapsulation techniques such as Frame Relay, High-Level Data Link Control (HDLC), LAPB-TA, and X.25 could support only one ISDN B-channel connection over the entire link. HDLC and PPP could support multiple B channels, but the entire ISDN link needed to use the same encapsulation. Dynamic multiple encapsulations allow incoming calls over ISDN to be assigned encapsulation type based on calling line identification (CLID) or Dialed Number Identification Service (DNIS). With dynamic multiple encapsulations, once CLID binding is completed, the topmost interface is always used for all configuration and data structures. The ISDN B channel becomes a forwarding device, and the configuration on the D channel is ignored, thereby allowing the different encapsulation types and per-user configurations.

To configure an interface for those networks, perform the tasks in the following sections:

- Configuring the Interface for Sending Traffic over a Frame Relay Network (As Required)
- Configuring the Interface for Sending Traffic over an X.25 Network (As Required)
- Configuring the Interface for Sending Traffic over a LAPB Network (As Required)

Configuring the Interface for Sending Traffic over a Frame Relay Network

Access to Frame Relay networks is now available through dialup connections and leased lines. Dialup connectivity allows Frame Relay networks to be extended to sites that do not generate enough traffic to justify leased lines, and also allows a Frame Relay network to back up another network or point-to-point line.

DDR over Frame Relay is supported for synchronous serial and ISDN interfaces and for rotary groups, and is available for in-band, DTR, and ISDN dialers.

Frame Relay supports multiple permanent virtual circuit (PVC) connections over the same serial interface or ISDN B channel, but only one *physical* interface can be used (dialed, connected, and active) in a rotary group or with ISDN.

Dynamic multiple encapsulations support the following Frame Relay features:

- Frame Relay RTP Header Compression (RFC 1889)
- Frame Relay TCP/IP Header Compression
- Legacy DDR over Frame Relay
- Frame Relay Interface/Subinterface Backup

Dynamic multiple encapsulations support at least four Frame Relay PVCs on either dialer interfaces or dialer subinterfaces.

Note Frame Relay encapsulations in the dynamic multiple encapsulations feature do not support IETF or Cisco Encapsulation for IBM Systems Network Architecture (SNA). Frame Relay for SNA support is not applicable.

Configuration Restrictions

The following restrictions apply to DDR used over Frame Relay:

- Frame Relay is not available for asynchronous dialers.
- The Frame Relay dynamic multiple encapsulations does *not* provide bidirectional support.
- With the dynamic multiple encapsulations, there is no process switching for Frame Relay packets; these packets are always fast switched.
- Like HDLC, LAPB, X.25 and Frame Relay do not provide authentication. However, ISDN dialers can offer some authentication through the caller ID feature.
- Only one ISDN B channel can be dialed at any one time. When configuring a rotary group, you can use only one serial interface.

Note Frame Relay subinterfaces work the same on dialup connections as they do on leased lines.

Configuration Overview

No new commands are required to support DDR over Frame Relay. In general, you configure Frame Relay and configure DDR. In general, to configure an interface for DDR over Frame Relay, perform the following tasks:

- Specify the interface.
- Specify the protocol identifiers for the interface.

  For example, enter the IP address and mask, the IPX network number, and the AppleTalk cable range and zone.
- Configure Frame Relay, as described in the chapter "Configuring Frame Relay" in the *Cisco IOS Wide-Area Networking Configuration Guide*.

  As a minimum, you must enable Frame Relay encapsulation and decide whether you need to do static or dynamic address mapping. If you decide to do dynamic mapping, you need not enter a command because Inverse ARP is enabled by default. If you decide to do static mapping, you must enter Frame Relay mapping commands.

  You can then configure various options as needed for your Frame Relay network topology.
- Configure DDR.

  At a minimum, you must decide and configure the interface for outgoing calls only, incoming calls only, or both outgoing and incoming calls.

  You can also configure DDR for your routed protocols (as specified in the chapter "Preparing to Configure DDR") and for snapshot routing (as specified in the chapter "Configuring Snapshot Routing" later in this publication). You can also customize DDR on your router or access server (as described in the "Customizing the Interface Settings" section later in this chapter).

For examples of configuring various interfaces for DDR over Frame Relay, see the section "Frame Relay Support Examples" later in this chapter.

Configuring the Interface for Sending Traffic over an X.25 Network

X.25 interfaces can now be configured to support DDR. Synchronous serial and ISDN interfaces on Cisco routers and access servers can be configured for X.25 addresses, X.25 encapsulation, and mapping of protocol addresses to the X.25 address of a remote host. In-band, DTR, and ISDN dialers can be configured to support X.25 encapsulation, but rotary groups cannot.

Remember that for ISDN interfaces, once CLID binding is completed, the topmost interface is always used for all configuration and data structures. The ISDN B channel becomes a forwarding device, and the configuration on the D channel is ignored, thereby allowing the different encapsulation types and per-user configurations. For X.25 encapsulations, the configurations reside on the dialer profile. The Dynamic Multiple Encapsulations feature provides support for packet assembler/disassembler (PAD) traffic and X.25 encapsulated and switched packets.

To configure an interface to support X.25 and DDR, use the following X.25-specific commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation x25 [dte | dce] [ietf]`	Configures the interface to use X.25 encapsulation.
Step 2	`Router(config-if)# x25 address x.121-address`	Assigns an X.25 address to the interface.
Step 3	`Router(config-if)# x25 map protocol address [protocol2 address2 [...[protocol9 address9]]] x.121-address [option]`	Sets up the LAN protocols-to-remote host address mapping.

The order of DDR and X.25 configuration tasks is not critical; you can configure DDR before or after X.25, and you can even mix the DDR and X.25 commands.

For an example of configuring an interface for X.25 encapsulation and then completing the DDR configuration, see the section "X.25 Support Configuration Example" later in this chapter.

Configuring the Interface for Sending Traffic over a LAPB Network

DDR over serial lines now supports LAPB encapsulation, in addition to the previously supported PPP, HDLC, and X.25 encapsulations.

LAPB encapsulation is supported on synchronous serial, ISDN, and dialer rotary group interfaces, but not on asynchronous dialers.

Because the default encapsulation is HDLC, you must explicitly configure LAPB encapsulation. To configure an interface to support LAPB encapsulation and DDR, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation lapb [dte | dce] [multi | protocol]`	Specifies LAPB encapsulation.

For more information about the serial connections on which LAPB encapsulation is appropriate, see the **encapsulation lapb** command in the chapter "X.25 and LAPB Commands" in the *Cisco IOS Wide-Area Networking Command Reference*, Release 12.2.

For an example of configuring an interface for DDR over LAPB, see the section "X.25 Support Configuration Example" later in this chapter.

Monitoring DDR Connections

To monitor DDR connections and snapshot routing, use the following commands in privileged EXEC mode:

Command	Purpose
`Router# show dialer [interface type number]`	Displays general diagnostics about the DDR interface.
`Router# show dialer map`	Displays current dialer maps, next-hop protocol addresses, user names, and the interfaces on which they are configured.
`Router# show interfaces bri 0`	Displays information about the ISDN interface.
`Router# show ipx interface [type number]`	Displays status about the IPX interface.
`Router# show ipx traffic`	Displays information about the IPX packets sent by the router or access server, including watchdog counters.
`Router# show appletalk traffic`	Displays information about the AppleTalk packets sent by the router or access server.
`Router# show vines traffic`	Displays information about the Banyan VINES packets sent by the router or access server.
`Router# show decnet traffic`	Displays information about the DECnet packets sent by the router or access server.
`Router# show xns traffic`	Displays information about the XNS packets sent by the router or access server.
`Router# clear dialer`	Clears the values of the general diagnostic statistics.

Configuration Examples for Legacy DDR Hub

The following sections provide various DDR configuration examples:

- Transparent Bridging over DDR Examples
- DDR Configuration in an IP Environment Example
- AppleTalk Configuration Example
- Banyan VINES Configuration Example
- DECnet Configuration Example
- ISO CLNS Configuration Example
- XNS Configuration Example
- Hub-and-Spoke DDR for Asynchronous Interfaces and Authentication Example
- Single Site or Multiple Sites Dialing Configuration Example
- Multiple Destinations Configuration Example
- Dialer Interfaces and Dialer Rotary Groups Example
- DDR Configuration Using Dialer Interface and PPP Encapsulation Example
- Two-Way DDR with Authentication Example

Transparent Bridging over DDR Examples

The following two examples differ only in the packets that cause calls to be placed. The first example specifies by protocol (any bridge packet is permitted to cause a call to be made); the second example allows a finer granularity by specifying the Ethernet type codes of bridge packets.

The first example configures serial interface 1 for DDR bridging. Any bridge packet is permitted to cause a call to be placed.

```
no ip routing
!
interface Serial1
 no ip address
 encapsulation ppp
 dialer in-band
 dialer enable-timeout 3
 dialer map bridge name urk broadcast 8985
 dialer hold-queue 10
 dialer-group 1
 ppp authentication chap
 bridge-group 1
 pulse-time 1
!
dialer-list 1 protocol bridge permit
bridge 1 protocol ieee
bridge 1 hello 10
```

The second example also configures the serial interface 1 for DDR bridging. However, this example includes an **access-list** command that specifies the Ethernet type codes that can cause calls to be placed and a **dialer list protocol list** command that refers to the specified access list.

```
no ip routing
!
interface Serial1
 no ip address
 encapsulation ppp
 dialer in-band
 dialer enable-timeout 3
 dialer map bridge name urk broadcast 8985
 dialer hold-queue 10
 dialer-group 1
 ppp authentication chap
 bridge-group 1
 pulse-time 1
!
access-list 200 permit 0x0800 0xFFF8
!
dialer-list 1 protocol bridge list 200
bridge 1 protocol ieee
bridge 1 hello 10
```

DDR Configuration in an IP Environment Example

The following example shows how to configure DDR to call one site from a synchronous serial interface in an IP environment. You could use the same configuration on an asynchronous serial interface by changing the **interface serial 1** command to specify an asynchronous interface (for example, **interface async 0**).

```
interface serial 1
 ip address 172.18.126.1 255.255.255.0
 dialer in-band
 dialer idle-timeout 600
 dialer string 5551234
 pulse-time 1
! The next command adds this interface to the dialer access group defined with
! the dialer-list command.
 dialer-group 1
!
! The first access list statement, below, specifies that IGRP updates are not
! interesting packets. The second access-list statement specifies that all
! other IP traffic such as Ping, Telnet, or any other IP packet is interesting.
! The dialer-list command then creates dialer access group 1 and states that
! access list 101 is to be used to classify packets as interesting or
! uninteresting. The ip route commands specify that there is a route to network
! 172.18.29.0 and to network 172.18.1.0 via 172.18.126.2. This means that
! several destination networks are available through a router that is dialed
! from interface serial 1.
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 list 101
ip route 172.18.29.0 172.18.126.2
ip route 172.18.1.0 172.18.126.2
ip local pool dialin 10.102.126.2 10.102.126.254
```

With many modems, the **pulse-time** command must be used so that DTR is dropped for enough time to allow the modem to disconnect.

AppleTalk Configuration Example

The following example configures DDR for AppleTalk access using an ISDN BRI. Two access lists are defined: one for IP and Interior Gateway Routing Protocol (IGRP) and one for AppleTalk. AppleTalk packets from network 2141 only (except broadcast packets) can initiate calls.

```
interface BRI0
 ip address 172.16.20.107 255.255.255.0
 encapsulation ppp
 appletalk cable-range 2141-2141 2141.65
 appletalk zone SCruz-Eng
 no appletalk send-rtmps
 dialer map ip 172.16.20.106 broadcast 1879
 dialer map appletalk 2141.66 broadcast 1879
 dialer-group 1
!
access-list 101 deny   igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 601 permit cable-range 2141-2141 broadcast-deny
access-list 601 deny other-access
!
dialer-list 1 list 101
dialer-list 1 list 601
```

Banyan VINES Configuration Example

The following example configures a router for VINES and IP DDR with in-band dialing. The VINES access list does not allow RTP routing updates to place a call, but any other data packet is interesting.

```
vines routing BBBBBBBB:0001
!
hostname RouterA
!
username RouterB password 7 030752180500
username RouterC password 7 00071A150754
!
interface serial 0
 ip address 172.18.170.19 255.255.255.0
 encapsulation ppp
 vines metrics 10
 vines neighbor AAAAAAAA:0001 0
 dialer in-band
 dialer map ip 172.18.170.151 name RouterB broadcast 4155551234
 dialer map vines AAAAAAAA:0001 name RouterC broadcast 4155551212
 dialer-group 1
 ppp authentication chap
 pulse-time 1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
vines access-list 107 deny RTP 00000000:0000 FFFFFFFF:FFFF 00000000:0000 FFFFFFFF:FFFF
vines access-list 107 permit IP 00000000:0000 FFFFFFFF:FFFF 00000000:0000 FFFFFFFF:FFFF
!
dialer-list 1 protocol ip list 101
dialer-list 1 protocol vines list 107
```

DECnet Configuration Example

The following example configures a router for DECnet DDR with in-band dialing:

```
decnet routing 10.19
username RouterB password 7 030752180531
!
interface serial 0
 no ip address
 decnet cost 10
 encapsulation ppp
 dialer in-band
 dialer map decnet 10.151 name RouterB broadcast 4155551212
 dialer-group 1
 ppp authentication chap
 pulse-time 1
!
access-list 301 permit 10.0 0.1023 0.0 63.1023
dialer-list 1 protocol decnet list 301
```

ISO CLNS Configuration Example

The following example configures a router for International Organization for Standardization Connectionless Network Service (ISO CLNS) DDR with in-band dialing:

```
username RouterB password 7 111C140B0E
clns net 47.0004.0001.0000.0c00.2222.00
clns routing
clns filter-set ddrline permit 47.0004.0001....
!
interface serial 0
 no ip address
 encapsulation ppp
 dialer in-band
 dialer map clns 47.0004.0001.0000.0c00.1111.00 name RouterB broadcast 1212
 dialer-group 1
 ppp authentication chap
 clns enable
 pulse-time 1
!
clns route default serial 0
dialer-list 1 protocol clns list ddrline
```

XNS Configuration Example

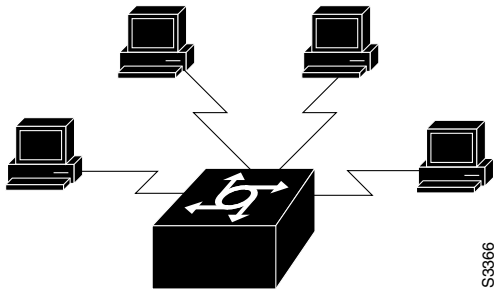
The following example configures a router for XNS DDR with in-band dialing. The access lists deny broadcast traffic to any host on any network, but allow all other traffic.

```
xns routing 0000.0c01.d8dd
username RouterB password 7 111B210A0F
interface serial 0
 no ip address
 encapsulation ppp
 xns network 10
 dialer in-band
 dialer map xns 10.0000.0c01.d877 name RouterB broadcast 4155551212
 dialer-group 1
 ppp authentication chap
 pulse-time 1
access-list 400 deny -1 -1.ffff.ffff.ffff 0000.0000.0000
access-list 400 permit -1 10
dialer-list 1 protocol xns list 400
```

Hub-and-Spoke DDR for Asynchronous Interfaces and Authentication Example

You can set up DDR to provide service to multiple remote sites. In a hub-and-spoke configuration, you can use a generic configuration script to set up each remote connection. Figure 54 illustrates a typical hub-and-spoke configuration.

Figure 54 Hub-and-Spoke DDR Configuration



The examples in the following sections show how to create this configuration.

Spoke Topology Configuration

The following commands are executed on the spoke side of the connection. (A different "spoke" password must be specified for each remote client.) The configuration provides authentication by identifying a password that must be provided on each end of the connection.

```
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 async mode dedicated
 async default ip address 172.19.45.1
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.1 name hub system-script hub 1234
 dialer map ip 172.30.45.255 name hub system-script hub 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 172.30.45.1
 ip default-network 172.30.0.0
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
 chat-script hub "" "" name: spoke1 word" <spoke1-passwd> PPP
 dialer-list 1 protocol ip permit
!
username hub password <spoke1-passwd>
!
router igrp 109
 network 172.30.0.0
 passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Hub Router Configuration

The following commands are executed on the local side of the connection—the hub router. The commands configure the server for communication with three clients and provide authentication by identifying a unique password for each "spoke" in the hub-and-spoke configuration.

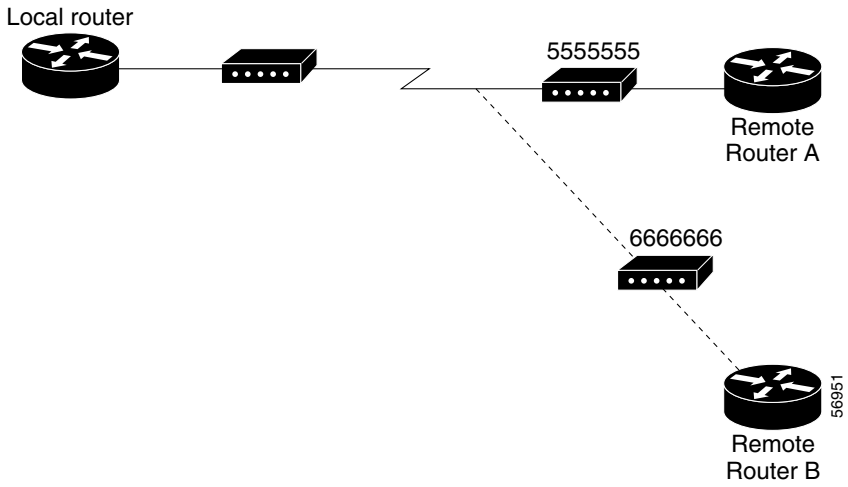
```
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode interactive
 async dynamic address
 dialer rotary-group 1
!
interface async 8
 async mode interactive
 async dynamic address
 dialer rotary-group 1
!
interface dialer 1
 ip address 172.30.45.2 255.255.255.0
 no ip split-horizon
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name spoke1 3333
 dialer map ip 172.30.45.2 name spoke2 4444
 dialer map ip 172.30.45.2 name spoke3 5555
 dialer map ip 172.30.45.255 name spoke1 3333
 dialer map ip 172.30.45.255 name spoke2 4444
 dialer map ip 172.30.45.255 name spoke3 5555
 dialer-group 1
!
ip route 172.30.44.0 255.255.255.0 172.30.45.2
ip route 172.30.44.0 255.255.255.0 172.30.45.3
ip route 172.30.44.0 255.255.255.0 172.30.45.4
 dialer-list 1 protocol ip list 101
 access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
 access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
username spoke1 password <spoke1-passwd>
username spoke2 password <spoke2-passwd>
username spoke3 password <spoke3-passwd>
username spoke1 autocommand ppp 172.30.45.2
username spoke2 autocommand ppp 172.30.45.3
username spoke3 autocommand ppp 172.30.45.4
!
router igrp 109
 network 172.30.0.0
 redistribute static
!
line 7
 login tacacs
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

The **redistribute static** command can be used to advertise static route information for DDR applications. Without this command, static routes to the hosts or network that the router can access with DDR will not be advertised to other routers with which the router is communicating. This behavior can block communication because some routes will not be known. See the **redistribute static ip** command, described in the chapter "IP Routing Protocol-Independent Commands" in the *Cisco IOS IP Command Reference, Volume 2 of 3: Routing Protocols*, Release 12.2.

Single Site or Multiple Sites Dialing Configuration Example

The following example is based on the configuration shown in Figure 55; the router receives a packet with a next hop address of 10.1.1.1.

Figure 55 Sample Dialer String or Dialer Map Configuration



If the interface on your router is configured to call a single site with phone number 5555555, it will send the packet to that site, assuming that the next hop address 10.1.1.1 indicates the same remote device as phone number 5555555. The **dialer string** command is used to specify the string (telephone number) to be called.

```
interface serial 1
 dialer in-band
 dialer string 5555555
```

If the interface is configured to dial multiple sites, the interface or dialer rotary group must be configured so that the correct phone number, 5555555, is mapped to the address 10.1.1.1. If this mapping is not configured, the interface or dialer rotary group does not know what phone number to call to deliver the packet to its correct destination, which is the address 10.1.1.1. In this way, a packet with a destination of 10.2.2.2 will not be sent to 5555555. The **dialer map** command is used to map next hop addresses to phone numbers.

```
interface serial 1
 dialer in-band
 dialer map ip 10.1.1.1 5555555
 dialer map ip 10.2.2.2 6666666
```

Multiple Destinations Configuration Example

The following example shows how to specify multiple destination numbers to dial for outgoing calls:

```
interface serial 1
 ip address 172.18.126.1 255.255.255.0
 dialer in-band
 dialer wait-for-carrier-time 100
 pulse-time 1
 dialer-group 1
 dialer map ip 172.18.126.10 5558899
```

```
 dialer map ip 172.18.126.15 5555555
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 255.255.255.255 0.0.0.0
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
dialer-list 1 protocol ip list 101
```

As in the "DDR Configuration in an IP Environment Example" section, a pulse time is assigned and a dialer access group specified.

The first **dialer map** command specifies that the number 555-8899 is to be dialed for IP packets with a *next-hop-address* value of 172.18.126.10. The second **dialer map** command then specifies that the number 5555555 will be called when an IP packet with a *next-hop-address* value of 172.18.126.15 is detected.

Dialer Interfaces and Dialer Rotary Groups Example

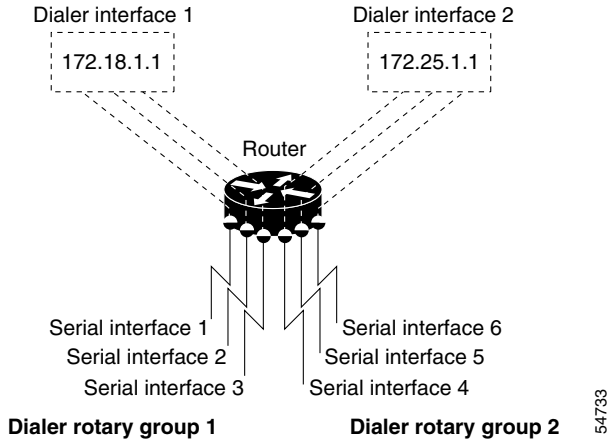
The following configuration places serial interfaces 1 and 2 into dialer rotary group 1, defined by the **interface dialer 1** command:

```
! PPP encapsulation is enabled for interface dialer 1.
interface dialer 1
 encapsulation ppp
 dialer in-band
 ip address 172.18.2.1 255.255.255.0
 ip address 172.18.2.1 255.255.255.0 secondary
! The first dialer map command allows remote site YYY and the central site to
! call each other. The second dialer map command, with no dialer string, allows
! remote site ZZZ to call the central site but the central site cannot call
! remote site ZZZ (no phone number).
!
 dialer map ip 172.18.2.5 name YYY 1415553434
 dialer map ip 172.18.2.55 name ZZZ
!
! The DTR pulse signals for three seconds on the interfaces in dialer group 1.
! This holds the DTR low so the modem can recognize that DTR has been dropped.
 pulse-time 3

! Serial interfaces 1 and 2 are placed in dialer rotary group 1. All the
! interface configuration commands (the encapsulation and dialer map commands
! shown earlier in this example) that applied to interface dialer 1 also apply
! to these interfaces.
interface serial 1
 dialer rotary-group 1
interface serial 2
 dialer rotary-group 1
```

DDR Configuration Using Dialer Interface and PPP Encapsulation Example

The following example shows a configuration for XXX, the local router shown in Figure 56. In this example, remote Routers YYY and ZZZ can call Router XXX. Router XXX has dialing information only for Router YYY and cannot call Router ZZZ.

Figure 56 DDR Configuration



Router XXX Configuration

```
username YYY password theirsystem
username ZZZ password thatsystem

! Create a dialer interface with PPP encapsulation and CHAP authentication.
interface dialer 1
 ip address 172.18.2.1 255.255.255.0
 ip address 172.24.4.1 255.255.255.0 secondary
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer group 1
! The first dialer map command indicates that calls between the remote site
! YYY and the central site will be placed at either end. The second dialer
! map command, with no dialer string, indicates that remote site ZZZ will call
! the central site but the central site will not call out.
 dialer map ip 172.18.2.5 name YYY 1415553434
 dialer map ip 172.24.4.5 name ZZZ
! The DTR pulse holds the DTR low for three seconds, so the modem can recognize
! that DTR has been dropped.
 pulse-time 3
!
! Place asynchronous serial interfaces 1 and 2 in dialer group 1. The interface commands
! applied to dialer group 1 (for example, PPP encapsulation and CHAP) apply to these
! interfaces.
!
interface async 1
 dialer rotary-group 1
interface async 2
 dialer rotary-group 1
```

Two-Way DDR with Authentication Example

You can set up two-way DDR with authentication in which both the client and server have dial-in access to each other. This configuration is demonstrated in the following two subsections.

Remote Configuration

The following commands are executed on the remote side of the connection. This configuration provides authentication by identifying a password that must be provided on each end of the connection.

```
username local password secret1
username remote password secret2
!
interface ethernet 0
 ip address 172.30.44.1 255.255.255.0
!
interface async 7
 ip address 172.30.45.2 255.255.255.0
 async mode dedicated
 async default ip address 172.30.45.1
 encapsulation ppp
 dialer in-band
 dialer string 1234
 dialer-group 1
!
ip route 172.30.43.0 255.255.255.0 async 7
ip default-network 172.30.0.0
chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
dialer-list 1 protocol ip permit
!
line 7
 no exec
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Local Configuration

The following commands are executed on the local side of the connection. As with the remote side configuration, this configuration provides authentication by identifying a password for each end of the connection.

```
username remote password secret1
username local password secret2
!
interface ethernet 0
 ip address 172.30.43.1 255.255.255.0
!
interface async 7
 async mode dedicated
 async default ip address 172.30.45.2
 dialer rotary-group 1
!
interface async 8
 async mode dedicated
 async default ip address 172.30.45.2
 dialer rotary-group 1
!
interface dialer 1
 ip address 172.30.45.2 255.255.255.0
 encapsulation ppp
 ppp authentication chap
 dialer in-band
 dialer map ip 172.30.45.2 name remote 4321
 dialer load-threshold 80
!
```

```
ip route 172.30.44.0 255.255.255.0 172.30.45.2
 chat-script generic ABORT BUSY ABORT NO ## AT OK ATDT\T TIMEOUT 30 CONNECT
!
router igrp 109
 network 172.30.0.0
 redistribute static
 passive-interface async 7
!
line 7
 modem InOut
 speed 38400
 flowcontrol hardware
 modem chat-script generic
```

Frame Relay Support Examples

The examples in this section present various combinations of interfaces, Frame Relay features, and DDR features.

Frame Relay Access with In-Band Dialing and Static Mapping

The following example configures a router for IP over Frame Relay using in-band dialing. A Frame Relay static map is used to associate the next hop protocol address to the DLCI. The dialer string allows dialing to only one destination.

```
interface Serial0
 ip address 10.1.1.1 255.255.255.0
 encapsulation frame-relay
 frame-relay map ip 10.1.1.2 100 broadcast
 dialer in-band
 dialer string 4155551212
 dialer-group 1
!
access-list 101 deny igrp any host 255.255.255.255
access-list 101 permit ip any any
!
dialer-list 1 protocol ip list 101
```

Frame Relay Access with ISDN Dialing and DDR Dynamic Maps

The following example shows a BRI interface configured for Frame Relay and for IP, Internet Protocol Exchange (IPX), and AppleTalk routing. No static maps are defined because this setup relies on Frame Relay Local Management Interface (LMI) signaling and Inverse ARP to determine the network addresses-to-DLCI mappings dynamically. (Because Frame Relay Inverse ARP is enabled by default, no command is required.)

```
interface BRI0
 ip address 10.1.1.1 255.255.255.0
 ipx network 100
 appletalk cable-range 100-100 100.1
 appletalk zone ISDN
 no appletalk send-rtmps
 encapsulation frame-relay IETF
 dialer map ip 10.1.1.2 broadcast 4155551212
 dialer map apple 100.2 broadcast 4155551212
 dialer map ipx 100.0000.0c05.33ed broadcast 4085551234
 dialer-group 1
!
```

```
access-list 101 deny igrp any host 255.255.255.255
access-list 101 permit ip any any
access-list 901 deny -1 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 457
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 457
access-list 901 permit -1
access-list 601 permit cable-range 100-100 broadcast-deny
access-list 601 deny other-access
!
dialer-list 1 protocol ip list 101
dialer-list 1 protocol novell list 901
dialer-list 1 protocol apple list 601
```

Frame Relay Access with ISDN Dialing and Subinterfaces

The following example shows a BRI interface configured for Frame Relay and for IP, IPX, and AppleTalk routing. Two logical subnets are used; a point-to-point subinterface and a multipoint subinterface are configured. Frame Relay Annex A (LMI type Q933a) and Inverse ARP are used for dynamic routing.

```
interface BRI0
 no ip address
 encapsulation frame-relay
 dialer string 4155551212
 dialer-group 1
 frame-relay lmi-type q933a
!
interface BRI0.1 multipoint
 ip address 10.1.100.1 255.255.255.0
 ipx network 100
 appletalk cable-range 100-100 100.1
 appletalk zone ISDN
 no appletalk send-rtmps
 frame-relay interface-dlci 100
 frame-relay interface-dlci 110
 frame-relay interface-dlci 120
!
interface BRI0.2 point-to-point
 ip address 10.1.200.1 255.255.255.0
 ipx network 200
 appletalk cable-range 200-200 200.1
 appletalk zone ISDN
 no appletalk send-rtmps
 frame-relay interface-dlci 200 broadcast IETF
!
access-list 101 deny igrp any host 255.255.255.255
access-list 101 permit ip any any
access-list 901 deny -1 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 457
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 452
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 453
access-list 901 deny -1 FFFFFFFF 0 FFFFFFFF 457
access-list 901 permit -1
access-list 601 permit cable-range 100-100 broadcast-deny
access-list 601 permit cable-range 200-200 broadcast-deny
access-list 601 deny other-access
```

```
dialer-list 1 protocol ip list 101
dialer-list 1 protocol novell list 901
dialer-list 1 protocol apple list 601
```

X.25 Support Configuration Example

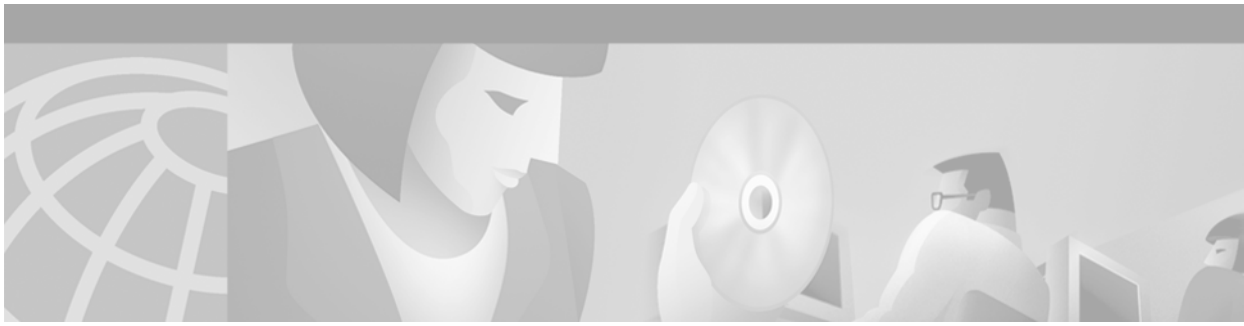
The following example configures a router to support X.25 and DTR dialing:

```
interface serial 0
 ip address 172.18.170.19 255.255.255.0
 encapsulation x25
 x25 address 12345
 x25 map ip 172.18.171.20 67890 broadcast
 dialer dtr
 dialer-group 1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 protocol ip list 101
```

LAPB Support Configuration Example

The following example configures a router for LAPB encapsulation and in-band dialing:

```
interface serial 0
 ip address 172.18.170.19 255.255.255.0
 encapsulation lapb
 dialer in-band
 dialer string 4155551212
 dialer-group 1
!
access-list 101 deny igrp 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
access-list 101 permit ip 0.0.0.0 255.255.255.255 0.0.0.0 255.255.255.255
!
dialer-list 1 protocol ip list 101
```



Configuring Peer-to-Peer DDR with Dialer Profiles

This chapter describes how to configure the Cisco IOS software for the Dialer Profiles feature implementation of dial-on-demand routing (DDR). It includes the following main sections:

- Dialer Profiles Overview
- How to Configure Dialer Profiles
- Monitoring and Maintaining Dialer Profile Connections
- Configuration Examples Dialer Profiles

For information about preparations for configuring dialer profiles, see the chapter "Preparing to Configure DDR" in this publication.

The Dialer Profiles feature is contrasted with legacy DDR. For information about legacy DDR, see the other chapters in the "Dial-on-Demand Routing" part of this publication.

For information about dial backup using dialer profiles, see the chapter "Configuring Dial Backup with Dialer Profiles" in this publication.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Dialer Profiles Overview

Dialer profiles allow the configuration of physical interfaces to be separated from the logical configuration required for a call, and they also allow the logical and physical configurations to be bound together dynamically on a per-call basis.

A dialer *profile* consists of the following elements:

- A *dialer interface* (a logical entity) configuration including one or more dial strings (each of which is used to reach one destination subnetwork)
- A *dialer map class* that defines all the characteristics for any call to the specified dial string
- An ordered *dialer pool* of physical interfaces to be used by the dialer interface

Note Dialer profiles support most routed protocols; however, International Organization for Standardization Connectionless Network Service (ISO CLNS) is not supported.

New Dialer Profile Model

In earlier releases of the Cisco IOS software, dialer profiles in the same dialer pool needed encapsulation-specific configuration information entered under both the dialer profile interface and the ISDN interface. If any conflict arose between the logical and the physical interfaces, the dialer profile failed to work.

In the new dialer profile model introduced by the Dynamic Multiple Encapsulations feature in Cisco IOS Release 12.1, the configuration on the ISDN interface is ignored and only the configuration on the profile interface is used, unless PPP name binding is used. Before a successful bind by CLID occurs, no encapsulation type and configuration are assumed or taken from the physical interfaces.

When PPP is used and a caller identification (CLID) bind fails, a dialer profile still can be matched by PPP name authentication. In the new dialer profile model, multiple attempts are made to find a matching profile.

The dialer profile software binds an incoming call on a physical dialer interface according to the following events, and in the order listed:

1. There is only one dialer profile configured to use the pool of which the physical interface is a member; this condition is the default bind. The physical interface must be a member of only this one pool. A default bind is possible only to a dialer profile when there are no **dialer caller** or **dialer called** commands configured on that profile.
2. The CLID matches what is configured in a **dialer caller** command on a dialer profile using a pool of which the physical interface is a member.
3. The DNIS that is presented matches what is configured in a **dialer called** command on a dialer profile using a pool of which the physical interface is a member.
4. If a bind has not yet occurred but the physical interface is configured for PPP encapsulation and CHAP or PAP authentication, and the CHAP or PAP name presented matches a **dialer remote-name** command configuration on a dialer profile using a pool of which the physical interface is a member, then the dialer profile software binds to that dialer profile.

If none of the above events are successful, the call is not answered. The call is also disconnected during any of the first three events when, after the bind occurs and the physical interface is configured for PPP encapsulation and CHAP or PAP authentication, the CHAP or PAP name presented does *not* match what is configured in a **dialer remote-name** command on the dialer profile that was bound to the call.

PPP encapsulation on an ISDN link is different from other encapsulation types because it runs on the B channel rather than the dialer profile interface. There are two possible configuration sources in a profile bind: the D and the dialer profile interfaces. Hence, a configuration conflict between the sources is possible. If a successful bind is accomplished by name authentication, the configuration used to bring PPP up is the one on the D interface. This is the name used to locate a dialer profile for the bind. The configuration on an ISDN interface goes under the D rather than a B channel, although B channels inherit the configuration from their D interface.

However, the configuration on this found dialer profile could be different from the one on the D interface. For example, the **ppp multilink** command is configured on the D interface, but not on the dialer profile interface. The actual per-user configuration is the one on the dialer profile interface. In this case, per-user configuration is not achieved unless link control protocol (LCP) and authentication are

renegotiated. Because PPP client software often does not accept renegotiation, this workaround is not acceptable. Therefore, the D interface configuration takes precedence over the dialer profile interface configuration. This is the only case where the configuration of the dialer profile is overruled.

Dialer Interface

A dialer interface configuration includes all settings needed to reach a specific destination subnetwork (and any networks reached through it). Multiple dial strings can be specified for the same dialer interface, each dial string being associated with a different dialer map class.

Dialer Map Class

The dialer map class defines all the characteristics for any call to the specified dial string. For example, the map class for one destination might specify a 56-kbps ISDN speed; the map class for a different destination might specify a 64-kbps ISDN speed.

Dialer Pool

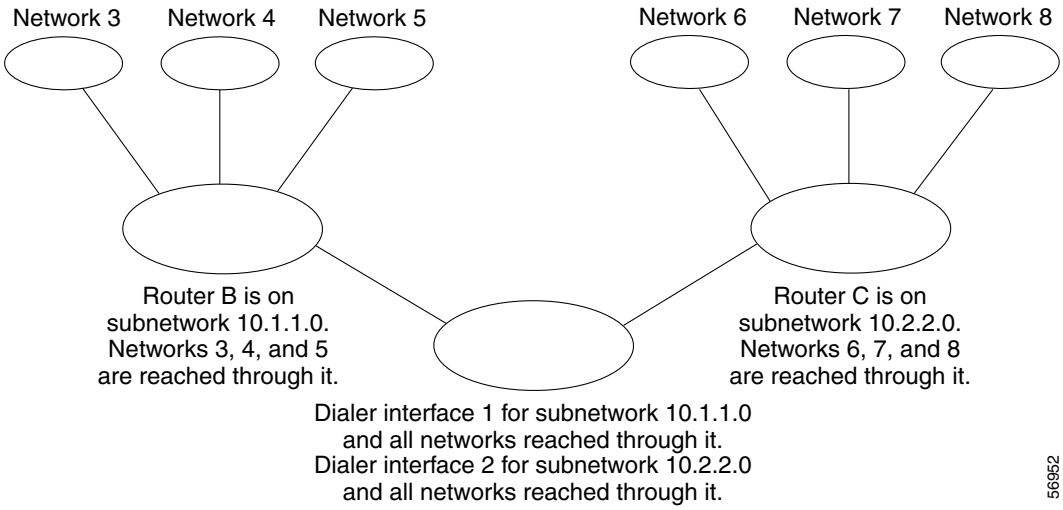
Each dialer interface uses a dialer pool, a pool of physical interfaces ordered on the basis of the priority assigned to each physical interface. A physical interface can belong to multiple dialer pools, contention being resolved by priority. ISDN BRI and PRI interfaces can set a limit on the minimum and maximum number of B channels reserved by any dialer pools. A channel reserved by a dialer pool remains idle until traffic is directed to the pool.

When dialer profiles are used to configure DDR, a physical interface has no configuration settings except encapsulation and the dialer pools with which the interface belongs.

Note The preceding paragraph has one exception: commands that apply before authentication is complete must be configured on the physical (or BRI or PRI) interface and not on the dialer profile. Dialer profiles do not copy PPP authentication commands (or LCP commands) to the physical interface.

Figure 57 shows a typical application of dialer profiles. Router A has dialer interface 1 for DDR with subnetwork 10.1.1.0, and dialer interface 2 for DDR with subnetwork 10.2.2.0. The IP address for dialer interface 1 is its address as a node in network 10.1.1.0; at the same time, that IP address serves as the IP address of the physical interfaces used by the dialer interface 1. Similarly, the IP address for dialer interface 2 is its address as a node in network 10.2.2.0.

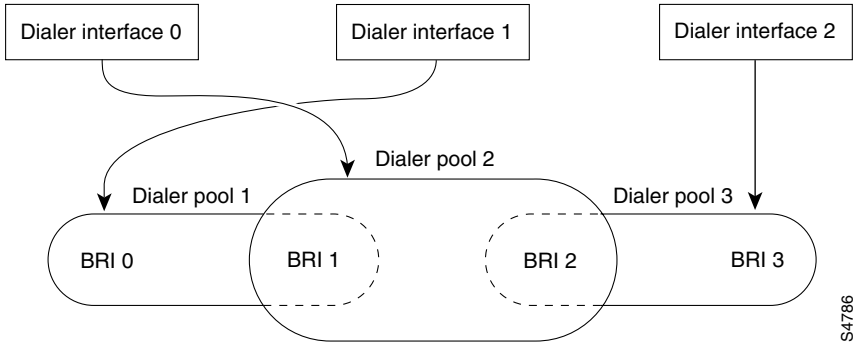
Figure 57 Typical Dialer Profiles Application



A dialer interface uses only one dialer pool. A physical interface, however, can be a member of one or many dialer pools, and a dialer pool can have several physical interfaces as members.

Figure 58 illustrates the relations among the concepts of dialer interface, dialer pool, and physical interfaces. Dialer interface 0 uses dialer pool 2. Physical interface BRI 1 belongs to dialer pool 2 and has a specific priority in the pool. Physical interface BRI 2 also belongs to dialer pool 2. Because contention is resolved on the basis of priority levels of the physical interfaces in the pool, BRI 1 and BRI 2 must be assigned different priorities in the pool. Perhaps BRI 1 is assigned priority 50 and BRI 2 is assigned priority 100 in dialer pool 2 (a priority of 100 is higher than a priority of 50). BRI 2 has a higher priority in the pool, and its calls will be placed first.

Figure 58 Relations Among Dialer Interfaces, Dialer Pools, and Physical Interfaces



How to Configure Dialer Profiles

To configure dialer profiles, perform the task in the following section:

- Configuring a Dialer Profile (Required)

The following tasks can be configured whether you use legacy DDR or dialer profiles. Perform these tasks as needed for your network:

- Configuring Dialer Profiles for Routed Protocols (As required)
- Configuring Dialer Profiles for Transparent Bridging (As required)

See the "Verifying the Dynamic Multiple Encapsulations Feature" section later in this chapter for tips on verifying that the feature is running in your network. See the "Configuration Examples Dialer Profiles" section at the end of this chapter for comprehensive configuration examples.

Configuring a Dialer Profile

To configure a dialer profile, perform the tasks in the following sections as required:

- Configuring a Dialer Interface (Required)
- Fancy Queueing and Traffic Shaping on Dialer Profile Interfaces (Optional)
- Configuring a Map Class (Optional)
- Configuring the Physical Interfaces (Required)

Configuring a Dialer Interface

Any number of dialer interfaces can be created for a router. Each dialer interface is the complete configuration for a destination subnetwork and any networks reached through it. The router on the destination subnetwork sends traffic on to the appropriate shadowed networks.

To configure a dialer interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Creates a dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address address mask`	Specifies the IP address and mask of the dialer interface as a node in the destination network to be called.
Step 3	`Router(config-if)# encapsulation type`	Specifies the encapsulation type.
Step 4	`Router(config-if)# dialer string dial-string class class-name`	Specifies the remote destination to call and the map class that defines characteristics for calls to this destination.
Step 5	`Router(config-if)# dialer pool number`	Specifies the dialing pool to use for calls to this destination.
Step 6	`Router(config-if)# dialer-group group-number`	Assigns the dialer interface to a dialer group.
Step 7	`Router(config-if)# dialer-list dialer-group protocol protocol-name {permit | deny | list access-list-number}`	Specifies an access list by list number or by protocol and list number to define the "interesting" packets that can trigger a call.

Fancy Queueing and Traffic Shaping on Dialer Profile Interfaces

In earlier releases of the Cisco IOS software, fancy queueing and traffic shaping were configured under the physical interfaces, therefore the same queueing or traffic shaping scheme needed to be applied to all users that were sharing the same ISDN link.

Beginning in Cisco IOS Release 12.1, you need only configure the queueing and traffic shaping schemes you desire on the dialer profile interface and the interface will take precedence over those configured on the ISDN B-channel interface. All the per-user encapsulation configuration has been moved to the dialer profile interfaces, separating it from hardware interfaces to make it dynamic and also to make per-user queueing and traffic shaping configuration possible.

Note Per-user fancy queueing and traffic shaping work with both process switching and fast switching in the new dialer profile model. However, Frame Relay Traffic Shaping (FRTS) is not supported on the new dialer profile model.

See the chapter "Policing and Shaping Overview" in the *Cisco IOS Quality of Service Solutions Configuration Guide* for more information about FRTS.

Configuring a Map Class

Map-class configuration is optional but allows you to specify different characteristics for different types of calls on a per-call-destination basis. For example, you can specify higher priority and a lower wait-for-carrier time for an ISDN-calls map class than for a modem-calls map class. You can also specify a different speed for some ISDN calls than for other ISDN calls.

A specific map class is tied to a specific call destination by the use of the map-class name in the **dialer-string** command with the **class** keyword.

To specify a map class and define its characteristics, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# map-class dialer classname`	Specifies a map class and begins map-class configuration mode.
Step 2	`Router(config-map-class)# dialer fast-idle seconds`	Specifies the fast idle timer value.
Step 3	`Router(config-map-class)# dialer idle-timeout seconds [inbound | either]`	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 4	`Router(config-map-class)# dialer wait-for-carrier-time seconds`	Specifies the length of time to wait for a carrier when dialing out to the dial string associated with the map class.
Step 5	`Router(config-map-class)# dialer isdn [speed speed] [spc]`	For ISDN only, specifies the bit rate used on the B channel associated with a specified map class or specifies that an ISDN semipermanent connection is to be used for calls associated with this map.

Note The **dialer idle-timeout** interface configuration command specifies the duration of time before an idle connection is disconnected. Previously, both inbound and outbound traffic would reset the dialer idle timer; now you can specify that only inbound traffic will reset the dialer idle timer.

Configuring the Physical Interfaces

To configure a physical interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the physical interface and begins interface configuration mode.
Step 2	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 3	`Router(config-if)# ppp authentication chap`	Specifies PPP Challenge Handshake Authentication Protocol (CHAP) authentication, if you also want to receive calls on this interface.

Step 4	`dialer pool-member number [priority priority]`
	
	`dialer pool-member number [priority priority] [min-link minimum] [max-link maximum]`
		Places the interface in a dialing pool and, optionally, assigns the interface a priority.
		
		For ISDN interfaces, you may also specify the minimum number of channels reserved and maximum number of channels used on this interface.
		
		The *minimum* value applies to outgoing calls only, and specifies the number of channels or interfaces reserved for dial out in that dialer pool; the channels remain idle when no calls are active. The *maximum* value applies to both incoming and outgoing calls and sets the total number of connections for a particular dialer pool member.

Step 5	`Router(config-if)# dialer pool-member number [priority priority]`
	or
	`Router(config-if)# dialer pool-member number [priority priority] [min-link minimum] [max-link maximum]`
		(Optional) Repeat Step 4 if you want to put the interface in additional dialing pools.

Repeat this procedure for additional physical interfaces that you want to use with dialer profiles.

Configuring Dialer Profiles for Routed Protocols

Both legacy DDR and dialer profiles support the following routed protocols: AppleTalk, Banyan VINES, DECnet, IP, Novell Internet Protocol Exchange (IPX), and Xerox Network System (XNS). To configure dialer profiles for a routed protocol, perform the tasks in the relevant section:

- Configuring Dialer Profiles for AppleTalk (As required)
- Configuring Dialer Profiles for Banyan VINES (As required)
- Configuring Dialer Profiles for DECnet (As required)

- Configuring Dialer Profiles for IP (As required)
- Configuring Dialer Profiles for Novell IPX (As required)
- Configuring XNS over DDR (As required)

Configuring Dialer Profiles for AppleTalk

To configure dialer profiles for AppleTalk, you specify AppleTalk access lists and then configure the dialer interface for dialer profiles, defining the dialer list to be used. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the section "Configuring a Dialer Interface" earlier in this chapter for more information about defining dialer lists.

Configuring Dialer Profiles for Banyan VINES

To configure DDR for Banyan VINES, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# vines access-list access-list-number {permit | deny} source source-mask1`	Specifies a VINES standard access list.
or	
`Router(config)# vines access-list access-list-number {permit | deny} source source-mask [destination] [destination-mask]`	Specifies a VINES extended access list.

After you specify VINES standard or extended access lists, configure the dialer interface for dialer profiles, defining the dialer list to be used. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the section "Configuring a Dialer Interface" earlier in this chapter for more information about defining dialer lists.

Note The Banyan VINES **neighbor** command is not supported for Link Access Procedure, Balanced (LAPB) and X.25 encapsulations.

Configuring Dialer Profiles for DECnet

To configure dial-on-demand routing (DDR) for DECnet, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {permit | deny} source source-mask1`	Specifies a DECnet standard access list.
or	
`Router(config)# access-list access-list-number {permit | deny} source source-mask [destination] [destination-mask]`	Specifies a DECnet extended access list.

After you specify DECnet standard or extended access lists, configure the dialer interface for dialer profiles, defining the dialer list to be used. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the section "Configuring a Dialer Interface" earlier in this chapter for more information about defining dialer lists.

You classify DECnet control packets, including hello packets and routing updates, using one or more of the following commands: **dialer-list protocol decnet_router-L1 permit**, **dialer-list protocol decnet_router-L2 permit**, and **dialer-list protocol decnet_node permit**.

Configuring Dialer Profiles for IP

To configure DDR for IP, use one of the following commands in global configuration mode:

Command	Purpose
`Router(config)# access-list access-list-number {deny | permit} source [source-mask]`	Specifies an IP standard access list.
or	
`Router(config)# access-list access-list-number {deny | permit} protocol source source-mask destination destination-mask [operator operand]`	Specifies an IP extended access list.

You can now also use simplified IP access lists that use the **any** keyword instead of the numeric forms of source and destination addresses and masks. Other forms of IP access lists are also available. For more information, see the chapter "IP Services Commands" in the *Cisco IOS IP Command Reference.*

To use dynamic routing where multiple remote sites communicate with each other through a central site, you might need to disable the IP split horizon feature. Split horizon applies to Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), and Enhanced IGRP. Depending on which routing protocol is configured, see the chapter "Configuring RIP," "Configuring IGRP," or "Configuring Enhanced IGRP" in this publication. Refer to the chapter "Configuring IP Routing Protocols" in the *Cisco IOS IP Configuration Guide* for more information.

Configuring Dialer Profiles for Novell IPX

On DDR links for Novell IPX, the link may come up often even when all client sessions are idle because the server sends watchdog or keepalive packets to all the clients approximately every 5 minutes. You can configure a local router or access server to idle out the DDR link and respond to the watchdog packets on behalf of the clients.

To modify the dialer profiles dialer interface configuration for Novell IPX, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no ipx route-cache`	Disables fast switching for IPX.

Step 2	`Router(config-if)# ipx watchdog-spoof`
	or
	`Router(config-if)# ipx spx-spoof`
		Enables IPX watchdog spoofing.
		Enables Sequenced Packet Exchange (SPX) keepalive spoofing.

Step 3	`Router(config-if)# ipx spx-idle-time delay-in-seconds`	Sets the idle time after which SPX keepalive spoofing begins.

Configuring XNS over DDR

To configure XNS for DDR, use one of the following commands in global configuration mode:

Command	Purpose

`Router(config)# access-list access-list-number {deny | permit} source-network[.source-address [source-address-mask]] [destination-network[.destination-address [destination-address-mask]]]`
or
`Router(config)# access-list access-list-number {deny | permit} protocol [source-network[.source-host [source-network-mask.]source-host-mask] source-socket [destination-network [.destination-host [destination-network-mask.destination-host-mask] destination-socket[/pep]]]`
	Specifies a standard XNS access list.
	Specifies an extended XNS access list.

After you specify an XNS access list, configure the dialer interface for dialer profiles, defining the dialer list to be used. Use the **dialer-list protocol** command to define permit or deny conditions for the entire protocol; for a finer granularity, use the **dialer-list protocol** command with the **list** keyword. See the section "Configuring a Dialer Interface" earlier in this chapter for more information about defining dialer lists.

Configuring Dialer Profiles for Transparent Bridging

The Cisco IOS software supports transparent bridging over both legacy DDR and dialer profiles, and it provides you some flexibility in controlling access and configuring the interface.

To configure dialer profiles for bridging, perform the tasks in the following sections:

- Defining the Protocols to Bridge (Required)
- Specifying the Bridging Protocol (Required)
- Controlling Access for Bridging (Required)
- Configuring an Interface for Bridging (Required)

Defining the Protocols to Bridge

IP packets are routed by default unless they are explicitly bridged; all others are bridged by default unless they are explicitly routed. To bridge IP packets, use the following command in global configuration mode:

Command	Purpose
`Router(config)# no ip routing`	Disables IP routing.

If you choose *not* to bridge another protocol, use the relevant command to enable routing of that protocol. For more information about tasks and commands, refer to the relevant chapter in the appropriate network protocol configuration guide, such as the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Specifying the Bridging Protocol

You must specify the type of spanning-tree bridging protocol to use and also identify a bridge group. To specify the spanning-tree protocol and a bridge group number, use the following command in global configuration mode:

Command	Purpose
`Router(config)# bridge` *bridge-group* `protocol {ieee | dec}`	Defines the type of spanning-tree protocol and identifies a bridge group.

The *bridge-group* number is used when you configure the interface and assign it to a bridge group. Packets are bridged only among members of the same bridge group.

Controlling Access for Bridging

You can control access by defining any transparent bridge packet as *interesting*, or you can use the finer granularity of controlling access by Ethernet type codes. To control access for DDR bridging, perform *one* of the following tasks:

- Permitting All Bridge Packets
- Controlling Bridging Access by Ethernet Type Codes

Note Spanning-tree bridge protocol data units (BPDUs) are always treated as *uninteresting*.

Permitting All Bridge Packets

To identify all transparent bridge packets as interesting, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer-list` *dialer-group* `protocol bridge permit`	Defines a dialer list that treats all transparent bridge packets as interesting.

Controlling Bridging Access by Ethernet Type Codes

To control access by Ethernet type codes, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number {permit | deny} type-code [mask]`	Identifies interesting packets by Ethernet type codes (access list numbers must be in the range 200 to 299).
Step 2	`Router(config)# dialer-list dialer-group protocol bridge list access-list-number`	Defines a dialer list for the specified access list.

For a table of some common Ethernet type codes, see the "Ethernet Type Codes" appendix in the *Cisco IOS Bridging and IBM Networking Command Reference*.

Configuring an Interface for Bridging

You can perform serial interfaces or ISDN interfaces for DDR bridging. To configure an interface for DDR bridging, complete all the tasks in the following sections:

- Specifying the Interface (Required)
- Configuring the Destination (Required)
- Assigning the Interface to a Bridge Group (Required)

Specifying the Interface

To specify the interface and enter interface configuration mode, use the following command in global configuration mode:

Command	Purpose
`Router(config)# interface type number`	Specifies the serial or ISDN interface and enters interface configuration mode.

Configuring the Destination

You can configure the destination by specifying either of the following:

- A dial string—for unauthenticated calls to a single site
- A dialer bridge map—when you want to use authentication

To configure the destination for bridging over a specified interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer string dial-string`	Configures the dial string to call.

Note You can define only one dialer bridge map for the interface. If you enter a different bridge map, the previous one is replaced immediately.

Assigning the Interface to a Bridge Group

Packets are bridged only among interfaces that belong to the same bridge group. To assign an interface to a bridge group, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# bridge-group bridge-group`	Assigns the specified interface to a bridge group.

Monitoring and Maintaining Dialer Profile Connections

To monitor DDR dialer profile connections, use any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show dialer interface`	Displays information for the interfaces configured for DDR dialer profiles.
`Router# show interfaces type number`	Displays statistics for configured interfaces. The output varies, depending on the network for which an interface has been configured.
`Router# show ipx interface [type number]`	Displays status about the IPX interface.
`Router# show ipx traffic`	Displays information about the IPX packets sent by the router or access server, including watchdog counters.
`Router# show appletalk traffic`	Displays information about the AppleTalk packets sent by the router or access server.
`Router# show vines traffic`	Displays information about the Banyan VINES packets sent by the router or access server.
`Router# show decnet traffic`	Displays information about the DECnet packets sent by the router or access server.
`Router# show xns traffic`	Displays information about the XNS packets sent by the router or access server.
`Router# clear dialer`	Clears the values of the general diagnostic statistics.

Configuration Examples Dialer Profiles

The following sections provide three comprehensive configuration examples:

- Dialer Profile with Inbound Traffic Filter Example
- Dialer Profile for Central Site with Multiple Remote Sites Example
- Dialer Profile for ISDN BRI Backing Up Two Leased Lines Example
- Dynamic Multiple Encapsulations over ISDN Example

Dialer Profile with Inbound Traffic Filter Example

The following example shows a Cisco 5200 series router that has enabled the **dialer idle-timeout** command with the **inbound** keyword. This command allows only inbound traffic that conforms to the dialer list to establish a connection and reset the dialer idle timer.

```
interface Serial0:23
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer pool-member 1 max-link 2
 isdn switch-type primary-5ess
 no cdp enable
 ppp authentication chap
!
interface Dialer0
 ip address 10.1.1.2 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 dialer remote-name 2610-2
 dialer idle-timeout 30 inbound
 dialer string 2481301
 dialer pool 1
 dialer-group 1
 no cdp enable
 ppp authentication chap
 ppp multilink
!
access-list 101 permit icmp any any
access-list 101 deny   ip any any
dialer-list 1 protocol ip list 101
```

Dialer Profile for Central Site with Multiple Remote Sites Example

The following example shows a central site that can place or receive calls from three remote sites over four ISDN BRI lines. Each remote site is on a different IP subnet and has different bandwidth requirements; therefore, three dialer interfaces and three dialer pools are defined.

```
! This is a dialer profile for reaching remote subnetwork 10.1.1.1.
interface Dialer1
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer remote-name Smalluser
 dialer string 4540
 dialer pool 3
 dialer-group 1

! This is a dialer profile for reaching remote subnetwork 10.2.2.2.
interface Dialer2
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name Mediumuser
 dialer string 5264540 class Eng
 dialer load-threshold 50 either
 dialer pool 1
 dialer-group 2

! This is a dialer profile for reaching remote subnetwork 10.3.3.3.
interface Dialer3
 ip address 10.3.3.3 255.255.255.0
```

```
 encapsulation ppp
 dialer remote-name Poweruser
 dialer string 4156884540 class Eng
 dialer hold-queue 10
 dialer load-threshold 80
 dialer pool 2
 dialer-group 2

! This map class ensures that these calls use an ISDN speed of 56 kbps.
map-class dialer Eng
 isdn speed 56

interface BRI0
 encapsulation PPP
! BRI 0 has a higher priority than BRI 1 in dialer pool 1.
 dialer pool-member 1 priority 100
 ppp authentication chap

interface BRI1
 encapsulation ppp
 dialer pool-member 1 priority 50
 dialer pool-member 2 priority 50
! BRI 1 has a reserved channel in dialer pool 3; the channel remains inactive
! until BRI 1 uses it to place calls.
 dialer pool-member 3 min-link 1
 ppp authentication chap

interface BRI2
 encapsulation ppp
! BRI 2 has a higher priority than BRI 1 in dialer pool 2.
 dialer pool-member 2 priority 100
 ppp authentication chap

interface BRI3
 encapsulation ppp
! BRI 3 has the highest priority in dialer pool 2.
 dialer pool-member 2 priority 150
 ppp authentication chap
```

Dialer Profile for ISDN BRI Backing Up Two Leased Lines Example

The following example shows the configuration of a site that backs up two leased lines using one BRI. Two dialer interfaces are defined. Each serial (leased line) interface is configured to use one of the dialer interfaces as a backup. Both of the dialer interfaces use BRI 0, and BRI 0 is a member of the two dialer pools. Thus, BRI 0 can back up two different serial interfaces and can make calls to two different sites.

```
interface dialer0
 ip unnumbered loopback0
 encapsulation ppp
 dialer remote-name Remote0
 dialer pool 1
 dialer string 5551212
 dialer-group 1

interface dialer1
 ip unnumbered loopback0
 encapsulation ppp
 dialer remote-name Remote1
 dialer pool 2
 dialer string 5551234
 dialer-group 1
```

```
interface bri 0
 encapsulation PPP
 dialer pool-member 1
 dialer pool-member 2
 ppp authentication chap

interface serial 0
 ip unnumbered loopback0
 backup interface dialer0
 backup delay 5 10

interface serial 1
 ip unnumbered loopback0
 backup interface dialer1
 backup delay 5 10
```

Dynamic Multiple Encapsulations over ISDN Example

The following example shows a network access server named NAS1 with dialer profiles and LAPB, X.25, and PPP encapsulations configured. Although the BRI0 D interface uses X.25 encapsulation, the actual encapsulations running over the ISDN B channels are determined by the encapsulations configured on the profile interfaces bound to them.

When an ISDN B channel connects to remote user RU2 using CLID 60043, Dialer1 is bound to this ISDN B channel by CLID binding. The protocol used is PPP; the X.25 configuration on the D interface has no effect. Because the **ppp authentication chap** command is configured, even though the binding is done by CLID, PPP authentication is still performed over the name RU2 before the protocol is allowed to proceed.

The Dialer2 interface uses DNIS-plus-ISDN-subaddress binding and is bound to a B channel with an incoming call with DNIS 60045 and ISDN subaddress 12345. Also note that the High-Level Data Link Control (HDLC) encapsulation has no username associated. It is no longer necessary to configure the **dialer remote-name** command, as in the previous dialer profile model.

When there is an ISDN B-channel connection to remote user RU1 using CLID 60036, LAPB encapsulation will run on this connection once CLID binding to Dialer0 takes place. This connection will operate as a standalone link independent of other activities over other ISDN B channels.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
service udp-small-servers
service tcp-small-servers
!
virtual-profile virtual-template 1
virtual-profile aaa
!
hostname NAS1
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable secret 5 $1$0Ced$YYJJl2p8f94lc/.JSgw8n1
enable password 7 153D19270D2E
!
username RU1 password 7 11260B2E1E16
username RU2 password 7 09635C221001
no ip domain-lookup
```

```
ip domain-name cisco.com
ip name-server 192.168.30.32
ip name-server 172.16.2.132
isdn switch-type basic-5ess
!
interface Virtual-Template 1
encapsulation ppp
ppp authentication chap
!
interface Ethernet0
 ip address 172.21.17.11 255.255.255.0
 no ip mroute-cache
 no cdp enable
!
interface Serial0
 ip address 10.2.2.1 255.0.0.0
 shutdown
 clock rate 56000
 ppp authentication chap
!
interface Serial1
 ip address 10.0.0.1 255.0.0.0
 shutdown
!
interface BRI0
 description PBX 60035
 no ip address
 encapsulation x25
 no ip mroute-cache
 no keepalive
 dialer pool-member 1
 dialer pool-member 2
!
interface Dialer0
 ip address 10.1.1.1 255.0.0.0
 encapsulation lapb dce multi
 no ip route-cache
 no ip mroute-cache
 no keepalive
 dialer remote-name RU1
 dialer idle-timeout 300
 dialer string 60036
 dialer caller 60036
 dialer pool 1
 dialer-group 1
 no fair-queue
!
interface Dialer1
 ip address 10.1.1.1 255.0.0.0
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer remote-name RU2
 dialer string 60043
 dialer caller 60043
 dialer pool 2
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap
!
interface Dialer2
 ip address 10.1.1.1 255.0.0.0
 encapsulation hdlc
```

```
 dialer called 60045:12345
 dialer pool 1
 dialer-group 1
 fair-queue
 !
radius-server host 172.19.61.87
radius-server key foobar
snmp-server community public RO
!
line con 0
 exec-timeout 0 0
line aux 0
 transport input all
line vty 0 4
 password 7 10611B320C13
 login
!
end
```

Verifying the Dynamic Multiple Encapsulations Feature

To see statistics on each physical interface bound to the dialer interface, and to verify dialer interfaces configured for binding, use the **show interfaces** EXEC command. Look for the reports "Bound to:" and "Interface is bound to..." while remembering that this feature applies only to ISDN.

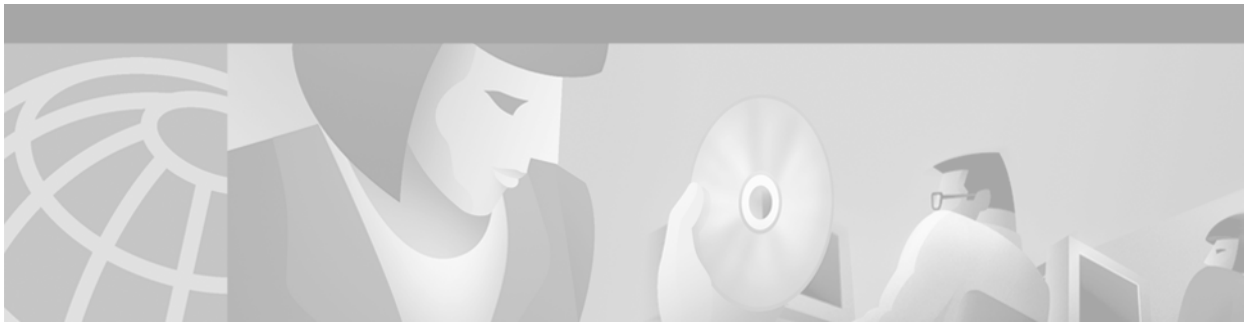
```
Router# show interfaces dialer0

Dialer0 is up, line protocol is up
  Hardware is Unknown
  Internet address is 10.1.1.2/8
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  DTR is pulsed for 1 seconds on reset
  Interface is bound to BRI0:1
  Last input 00:00:38, output never, output hang never
  Last clearing of "show interface" counters 00:05:36
  Queueing strategy: fifo
  Output queue 0/40, 0 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
     38 packets input, 4659 bytes
     34 packets output, 9952 bytes
Bound to:
BRI0:1 is up, line protocol is up
  Hardware is BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  Interface is bound to Dialer0 (Encapsulation PPP)
  LCP Open, multilink Open
  Last input 00:00:39, output 00:00:11, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 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
     78 packets input, 9317 bytes, 0 no buffer
     Received 65 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     93 packets output, 9864 bytes, 0 underruns
     0 output errors, 0 collisions, 7 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions
```

At the end of the Dialer0 display, the **show interfaces** command is executed on each physical interface bound to it.

In the next example, the physical interface is the B1 channel of the BRI0 link. This example also illustrates that the output under the B channel keeps all hardware counts that are not displayed under any logical or virtual access interface. The line in the report that states "Interface is bound to Dialer0 (Encapsulation LAPB)" indicates that this B interface is bound to the dialer 0 interface and that the encapsulation running over this connection is LAPB, not PPP, which is the encapsulation configured on the D interface and inherited by the B channel.

```
Router# show interfaces bri0:1

BRI0:1 is up, line protocol is up
  Hardware is BRI
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive not set
  Interface is bound to Dialer0 (Encapsulation LAPB)
  LCP Open, multilink Open
  Last input 00:00:31, output 00:00:03, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 1 packets/sec
  5 minute output rate 0 bits/sec, 1 packets/sec
     110 packets input, 13994 bytes, 0 no buffer
     Received 91 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     135 packets output, 14175 bytes, 0 underruns
     0 output errors, 0 collisions, 12 interface resets
     0 output buffer failures, 0 output buffers swapped out
     8 carrier transitions
```

Any protocol configuration and states should be displayed from the dialer 0 interface.



Configuring Snapshot Routing

This chapter describes how to configure snapshot routing. It includes the following main sections:

- Snapshot Routing Overview
- How to Configure Snapshot Routing
- Monitoring and Maintaining DDR Connections and Snapshot Routing
- Configuration Examples for Snapshot Routing

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the snapshot routing commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Snapshot Routing Overview

Snapshot routing enables a single router interface to call other routers during periods when the line protocol for the interface is up (these are called "active periods"). The router dials in to all configured locations during such active periods to get routes from all the remote locations.

The router can be configured to exchange routing updates each time the line protocol goes from "down" to "up" or from "dialer spoofing" to "fully up." The router can also be configured to dial the server router in the absence of regular traffic if the active period time expires.

Snapshot routing is useful in two command situations:

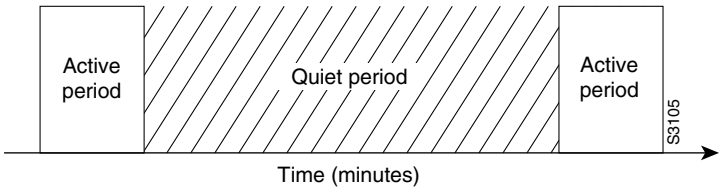
- Configuring static routes for dial-on-demand routing (DDR) interfaces
- Reducing the overhead of periodic updates sent by routing protocols to remote branch offices over a dedicated serial line

When configuring snapshot routing, you choose one router on the interface to be the client router and one or more other routers to be server routers. The client router determines the frequency at which routing information is exchanged between routers.

Routing information is exchanged during an active period. During the active period, a client router dials all the remote server routers for which it has a snapshot dialer map defined in order to get routes from all the remote locations. The server router provides information about routes to each client router that calls.

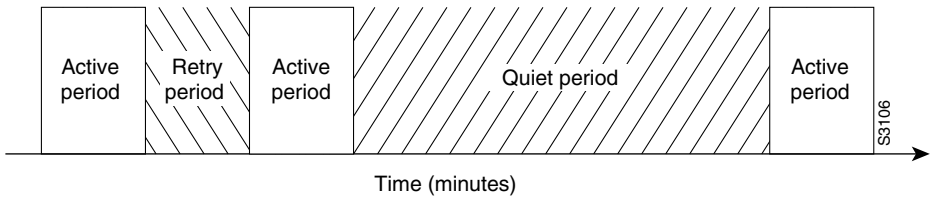
At the end of the active period, the router takes a snapshot of the entries in the routing table. These entries remain frozen during a quiet period. At the end of the quiet period, another active period starts during which routing information is again exchanged; see Figure 59.

Figure 59 Active and Quiet Periods in Snapshot Routing



When the router makes the transition from the quiet period to the active period, the line might not be available for a variety of reasons. For example, the line might be down or busy, or the permanent virtual circuit (PVC) might be down. If this happens, the router has to wait through another entire quiet period before it can update its routing table entries. This wait might be a problem if the quiet period is very long—for example, 12 hours. To avoid the need to wait through the quiet period, you can configure a retry period. If the line is not available when the quiet period ends, the router waits for the amount of time specified by the retry period and then makes the transition to an active period. See to Figure 60.

Figure 60 Retry Period in Snapshot Routing



The retry period is also useful in a dialup environment in which there are more remote sites than router interface lines that dial in to a PRI and want routing information from that interface. For example, a PRI has 23 DS0s available, but you might have 46 remote sites. In this situation, you would have more **dialer map** commands than available lines. The router will try the **dialer map** commands in order and will use the retry time for the lines that it cannot immediately access.

The following routed protocols support snapshot routing. Note that these are all distance-vector protocols.

- AppleTalk—Routing Table Maintenance Protocol (RTMP)
- Banyan VINES—Routing Table Protocol (RTP)
- IP—Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP)
- Internet Protocol Exchange (IPX)—RIP, Service Advertisement Protocol (SAP)

How to Configure Snapshot Routing

To configure snapshot routing, perform the tasks in the following sections:

- Configuring the Client Router (Required)
- Configuring the Server Router (Required)

You can also monitor and maintain interfaces configured for snapshot routing. For tips on maintaining your network with snapshot routing, see the section "Monitoring and Maintaining DDR Connections and Snapshot Routing" later in this chapter.

For an example of configuring snapshot routing, see the section "Configuration Examples for Snapshot Routing" at the end of this chapter.

Configuring the Client Router

To configure snapshot routing on the client router that is connected to a dedicated serial line, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies a serial interface.
Step 2	`Router(config-if)# snapshot client active-time quiet-time [suppress-statechange-updates] [dialer]`	Configures the client router.

To configure snapshot routing on the client router that is connected to an interface configured for DDR, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies a serial interface.
Step 2	`Router(config-if)# dialer rotary-group number`	Configures a dialer rotary group.
Step 3	`Router(config-if)# interface dialer number`	Specifies a dialer interface.
Step 4	`Router(config-if)# snapshot client active-time quiet-time [suppress-statechange-updates] [dialer]`	Configures the client router.
Step 5	`Router(config-if)# dialer map snapshot sequence-number dial-string`	Defines a dialer map.

Repeat these steps for each map you want to define. Maps must be provided for all the remote server routers that this client router is to call during each active period.

Because ISDN BRI and PRI automatically have rotary groups, you need not define a rotary group when configuring snapshot routing.

To configure snapshot routing on the client router over an interface configured for BRI or PRI, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri number`	Specifies a BRI interface.
Step 2	`Router(config-if)# snapshot client active-time quiet-time [suppress-statechange-updates] [dialer]`	Configures the client router.
Step 3	`Router(config-if)# dialer map snapshot sequence-number dial-string`	Defines a dialer map.

Configuring the Server Router

To configure snapshot routing on the server router that is connected to a dedicated serial line, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies a serial interface.
Step 2	`Router(config-if)# snapshot server active-time [dialer]`	Configures the server router.

To configure snapshot routing on the associated server router that is connected to an interface configured for DDR, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies a serial interface.
Step 2	`Router(config-if)# interface dialer number`	Specifies a dialer interface.
Step 3	`Router(config-if)# snapshot server active-time [dialer]`	Configures the server router.

The active period for the client router and its associated server routers should be the same.

Monitoring and Maintaining DDR Connections and Snapshot Routing

To monitor DDR connections and snapshot routing, use any of the following commands in privileged EXEC mode:

Command	Purpose
`Router# show dialer [interface type number]`	Displays general diagnostics about the DDR interface.
`Router# show interfaces bri 0`	Displays information about the ISDN interface.
`Router# clear snapshot quiet-time interface`	Terminates the snapshot routing quiet period on the client router within 2 minutes.
`Router# show snapshot [type number]`	Displays information about snapshot routing parameters.
`Router# clear dialer`	Clears the values of the general diagnostic statistics.

Configuration Examples for Snapshot Routing

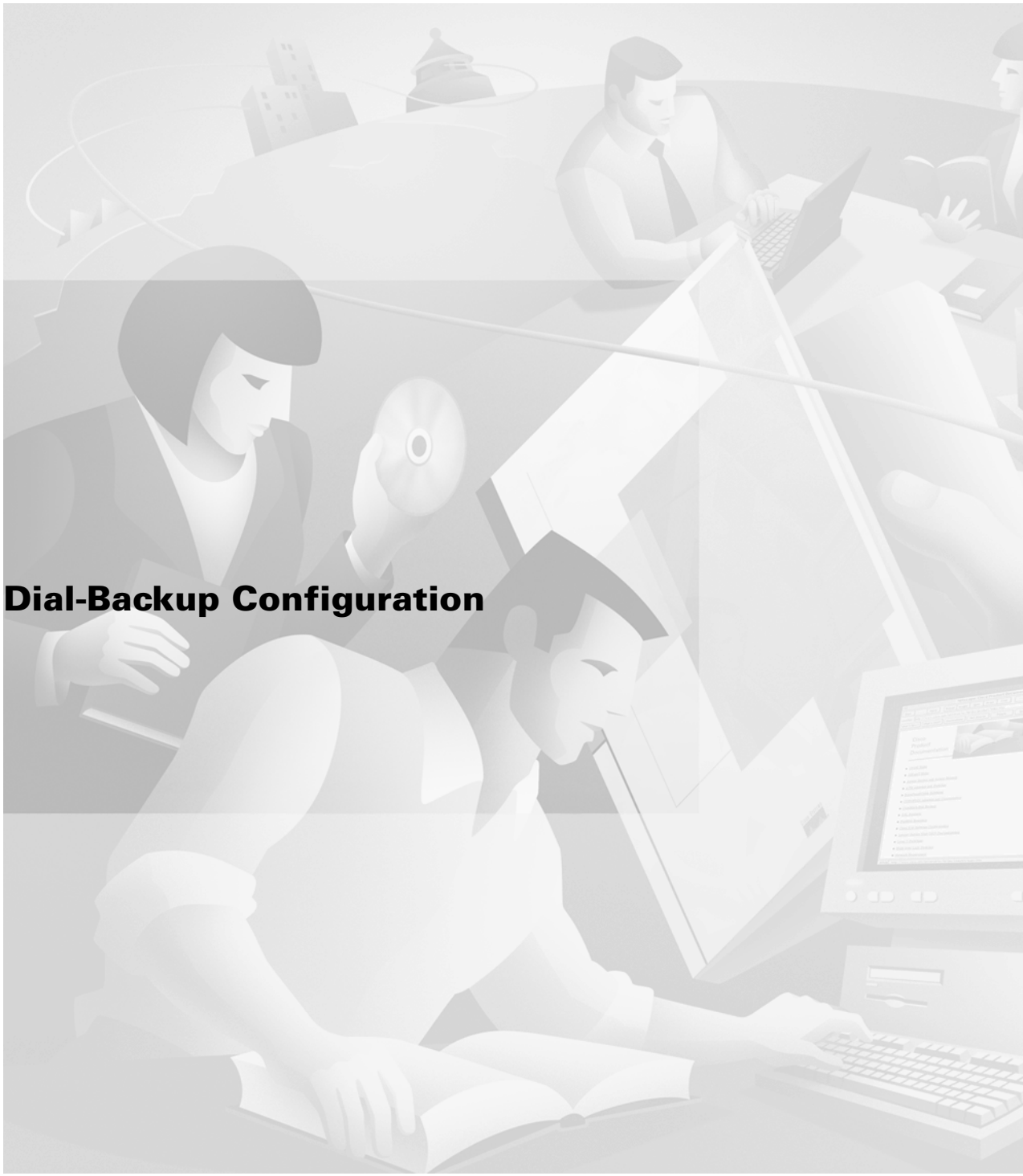
The following example configures snapshot routing on an interface configured for DDR on the client router. In this configuration, a single client router can call multiple server routers. The client router dials to all different locations during each active period to get routes from all those remote locations.

The absence of the **suppress-statechange-updates** keyword means that routing updates will be exchanged each time the line protocol goes from "down" to "up" or from "dialer spoofing" to "fully up." The **dialer** keyword on the **snapshot client** command allows the client router to dial the server router in the absence of regular traffic if the active period time expires.

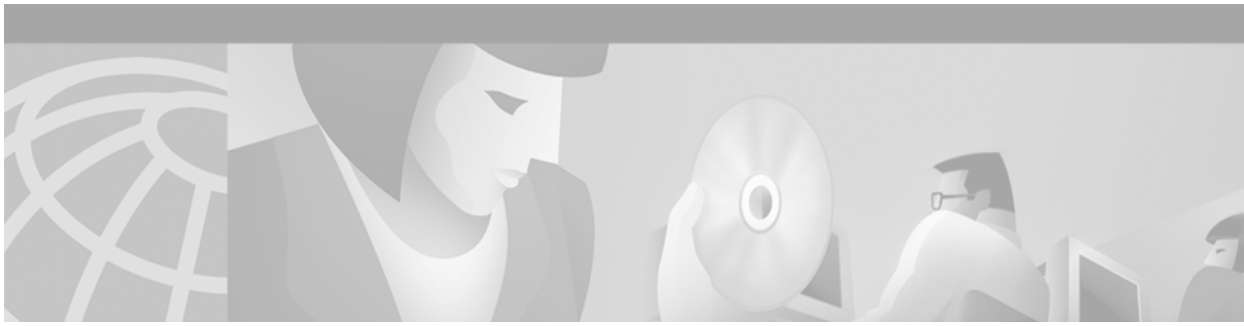
```
interface serial 0
 dialer rotary-group 3
!
interface dialer 3
 dialer in-band
 snapshot client 5 360 dialer

dialer map snapshot 2 4155556734
dialer map snapshot 3 7075558990
```

The following example configures the server router:

```
interface serial 2
 snapshot server 5 dialer
```



Dial-Backup Configuration



Configuring Dial Backup for Serial Lines

This chapter describes how to configure the primary interface to use the dial backup interface. It includes the following main sections:

- Backup Serial Interface Overview
- How to Configure Dial Backup
- Configuration Examples for Dial Backup for Serial Interfaces

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the dial backup commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Backup Serial Interface Overview

For a backup serial interface, an external DCE device, such as a modem attached to a circuit-switched service, must be connected to the backup serial interface. The external device must be capable of responding to a data terminal ready (DTR) Active signal by automatically dialing the preconfigured telephone number of the remote site.

A backup interface is an interface that stays idle until certain circumstances occur; then it is activated. A backup interface for a serial interface can be an ISDN interface or a different serial interface. A backup interface can be configured to be activated when any of the following three circumstances occurs:

- The primary line goes down.
- The load on the primary line reaches a certain threshold.
- The load on the primary line exceeds a specified threshold.

To configure a dial backup to a serial interface, you must configure the interface to use the dial backup interface, specify the conditions in which the backup interface will be activated, and then configure the dial-backup interface for dial-on-demand routing (DDR). The DDR configuration specifies the conditions and destinations for dial calls. The serial interface (often called the *primary* interface) might be configured for DDR or for Frame Relay or X.25 over a leased line, but the backup tasks are the same in all three cases.

Note Dial backup is also available using the Dialer Watch feature. Dialer Watch is based on routing characteristics instead of relying exclusively on interesting traffic conditions. For information about Dialer Watch, see the chapter "Configuring Dial Backup Using Dialer Watch" in this publication.

To configure a backup interface for a serial interface based on one of the conditions listed, complete the following general steps:

- Specify the interface and configure it as needed (for DDR, Frame Relay, or X.25). You can also specify and configure a Frame Relay subinterface.

  Refer to the chapters "Configuring Frame Relay" or "Configuring X.25" in the *Cisco IOS Wide-Area Networking Configuration Guide*. In this publication, see the chapter "Configuring Synchronous Serial Ports" and related chapters in the "Dial-on-Demand Routing" part for details.
- Configure the primary interface or subinterface by specifying the dial backup interface and the conditions for activating the backup interface, as described in this chapter.
- Configure the backup interface for DDR, as described in the "Dial-on-Demand Routing" part of this publication.

See the chapters "Configuring Legacy DDR Spokes" (for point-to-point legacy DDR connections) or "Configuring Legacy DDR Hubs" (for point-to-multipoint legacy DDR connections) in this publication. If you have configured dialer profiles instead of legacy DDR, see the chapter "Configuring Dial Backup with Dialer Profiles" in this publication for backup information.

How to Configure Dial Backup

You must decide whether to activate the backup interface when the primary line goes down, when the traffic load on the primary line exceeds the defined threshold, or both. The tasks you perform depend on your decision. Perform the tasks in the following sections to configure dial backup:

- Specifying the Backup Interface (Optional)
- Defining the Traffic Load Threshold (Optional)
- Defining Backup Line Delays (Optional)

Then configure the backup interface for DDR, so that calls are placed as needed. See the chapters in the "Dial-on-Demand Routing" part of this publication for more information.

For simple configuration examples, see the section "Configuration Examples for Dial Backup for Serial Interfaces" at the end of this chapter.

Specifying the Backup Interface

To specify a backup interface for a primary serial interface or subinterface, use one the following commands in interface configuration mode:

Command	Purpose

`Router(config-if)# backup interface type number`
or
Cisco 7500 series routers:
`Router(config-if)# backup interface type slot/port`
or
Cisco 7200 series routers:
`Router(config-if)# backup interface type slot/port-adapter/port`
	Selects a backup interface.

Note When you enter the **backup interface** command, the configured physical or logical interface will be forced to standby mode. When you use a BRI for a dial backup (with Legacy DDR), neither of the B channels can be used because the physical BRI interface is in standby mode. However, with dialer profiles, only the logical dialer interface is placed in standby mode and the physical interface (BRI) still can be used for other connections by making it a member of another pool.

When configured for legacy DDR, the backup interface can back up only one interface. For examples of selecting a backup line, see the sections "Dial Backup Using an Asynchronous Interface Example" and "Dial Backup Using DDR and ISDN Example" later in this chapter.

Defining the Traffic Load Threshold

You can configure dial backup to activate the secondary line based on the traffic load on the primary line. The software monitors the traffic load and computes a 5-minute moving average. If this average exceeds the value you set for the line, the secondary line is activated and, depending upon how the line is configured, some or all of the traffic will flow onto the secondary dialup line.

To define how much traffic should be handled at one time on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# backup load {enable-threshold | never} {disable-load | never}`	Defines the traffic load threshold as a percentage of the available bandwidth of the primary line.

Defining Backup Line Delays

You can configure a value that defines how much time should elapse before a secondary line status changes after a primary line status has changed. You can define two delays:

- A delay that applies after the primary line goes *down* but before the secondary line is activated
- A delay that applies after the primary line comes *up* but before the secondary line is deactivated

To define these delays, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# backup delay {enable-delay | never} {disable-delay | never}`	Defines backup line delays.

For examples of how to define backup line delays, see the sections "Dial Backup Using an Asynchronous Interface Example" and "Dial Backup Using DDR and ISDN Example" at the end of this chapter.

Configuration Examples for Dial Backup for Serial Interfaces

The following sections present examples of specifying the backup interface:

- Dial Backup Using an Asynchronous Interface Example
- Dial Backup Using DDR and ISDN Example

The following sections present examples of backup interfaces configured to be activated in three different circumstances:

- The load on the primary line reaches a certain threshold.
- The load on the primary line exceeds a specified threshold.
- The primary line goes down.

Dial Backup Using an Asynchronous Interface Example

The following is an example for dial backup using asynchronous interface 1, which is configured for DDR:

```
interface serial 0
 ip address 172.30.3.4 255.255.255.0
 backup interface async1
 backup delay 10 10
!
interface async 1
 ip address 172.30.3.5 255.255.255.0
 dialer in-band
 dialer string 5551212
 dialer-group 1
 async dynamic routing
dialer-list 1 protocol ip permit
chat-script sillyman "" "atdt 5551212" TIMEOUT 60 "CONNECT"
line 1
 modem chat-script sillyman
 modem inout
 speed 9600
```

Dial Backup Using DDR and ISDN Example

The following example shows how to use an ISDN interface to back up a serial interface.

Note When you use a BRI interface for dial backup, neither of the B channels can be used while the interface is in standby mode.

Interface BRI 0 is configured to make outgoing calls to one number. This is a legacy DDR spoke example.

```
interface serial 1
 backup delay 0 0
 backup interface bri 0
 ip address 10.2.3.4 255.255.255.0
!
interface bri 0
 ip address 10.2.3.5 255.255.255.0
 dialer string 5551212
 dialer-group 1
!
dialer-list 1 protocol ip permit
```

Note Dialing will occur only after a packet is received to be output on BRI 0. We recommend using the **dialer-list** command with the **protocol** and **permit** keywords specified to control access for dial backup. Using this form of access control specifies that all packets are interesting.

Dial Backup Service When the Primary Line Reaches Threshold Example

The following example configures the secondary line (serial 1) to be activated only when the load of the primary line reaches a certain threshold:

```
interface serial 0
 backup interface serial 1
 backup load 75 5
```

In this case, the secondary line will not be activated when the primary goes down. The secondary line will be activated when the load on the primary line is greater than 75 percent of the bandwidth of the primary line. The secondary line will then be brought down when the aggregate load between the primary and secondary lines fits within 5 percent of the primary bandwidth.

The same example on a Cisco 7500 series router would be as follows:

```
interface serial 1/1
 backup interface serial 2/2
 backup load 75 5
```

Dial Backup Service When the Primary Line Exceeds Threshold Example

The following example configures the secondary line (serial 1) to activate when the traffic threshold on the primary line exceeds 25 percent:

```
interface serial 0
 backup interface serial 1
 backup load 25 5
 backup delay 10 60
```

When the aggregate load of the primary and the secondary lines returns to within 5 percent of the primary bandwidth, the secondary line is deactivated. The secondary line waits 10 seconds after the primary goes down before activating and remains active for 60 seconds after the primary returns and becomes active again.

The same example on a Cisco 7500 series router would be as follows:

```
interface serial 1/0
 backup interface serial 2/0
 backup load 25 5
 backup delay 10 60
```

Dial Backup Service When the Primary Line Goes Down Example

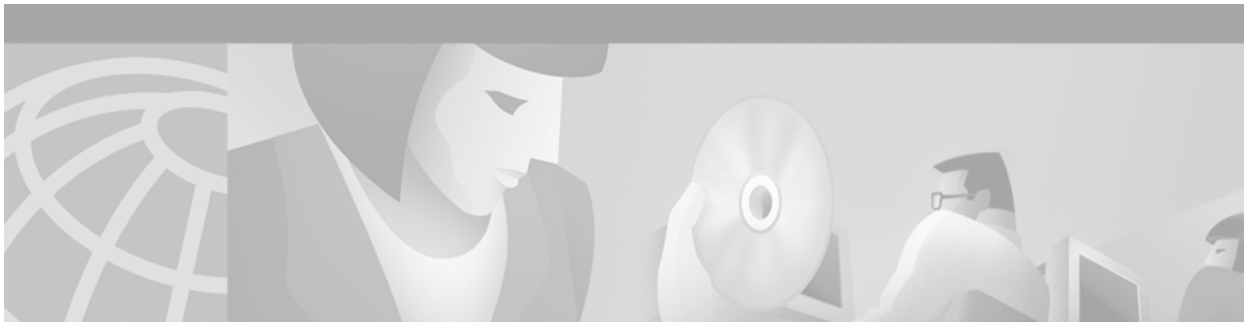
The following example configures the secondary line (serial 1) as a backup line that becomes active only when the primary line (serial 0) goes down. The backup line will not be activated because of load on the primary line.

```
interface serial 0
 backup interface serial 1
 backup delay 30 60
```

The backup line is configured to activate 30 seconds after the primary line goes down and to remain on for 60 seconds after the primary line is reactivated.

The same example on a Cisco 7500 series router would be as follows:

```
interface serial 1/1
 backup interface serial 2/0
 backup delay 30 60
```



Configuring Dial Backup with Dialer Profiles

This chapter describes how to configure dialer interfaces, which can be configured as the logical intermediary between one or more physical interfaces and another physical interface that is to function as backup. It includes the following main sections:

- Dial Backup with Dialer Profiles Overview
- How to Configure Dial Backup with Dialer Profiles
- Configuration Example of Dialer Profile for ISDN BRI Backing Up Two Leased Lines

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the dial backup commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Dial Backup with Dialer Profiles Overview

A backup interface is an interface that stays idle until certain circumstances occur; then it is activated. Dialer interfaces can be configured to use a specific dialing pool; in turn, physical interfaces can be configured to belong to the same dialing pool.

See the section "Configuration Example of Dialer Profile for ISDN BRI Backing Up Two Leased Lines" at the end of this chapter for a comprehensive example of a dial backup interface using dialer profiles. In the example, one BRI functions as backup to two serial lines and can make calls to two different destinations.

How to Configure Dial Backup with Dialer Profiles

To configure a dialer interface and a specific physical interface to function as backup to other physical interfaces, perform the tasks in the following sections:

- Configuring a Dialer Interface (Required)
- Configuring a Physical Interface to Function As Backup (Required)
- Configuring Interfaces to Use a Backup Interface (Required)

Configuring a Dialer Interface

To configure the dialer interface that will be used as an intermediary between a physical interface that will function as backup interface and the interfaces that will use the backup, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Creates a dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered loopback0`	Specifies IP unnumbered loopback.
Step 3	`Router(config-if)# encapsulation ppp`	Specifies PPP encapsulation.
Step 4	`Router(config-if)# dialer remote-name username`	Specifies the Challenge Handshake Authentication Protocol (CHAP) authentication name of the remote router.
Step 5	`Router(config-if)# dialer string dial-string`	Specifies the remote destination to call.
Step 6	`Router(config-if)# dialer pool number`	Specifies the dialing pool to use for calls to this destination.
Step 7	`Router(config-if)# dialer-group group-number`	Assigns the dialer interface to a dialer group.

Configuring a Physical Interface to Function As Backup

To configure the physical interface that is to function as backup, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 2	`Router(config-if)# encapsulation ppp`	Specifies PPP encapsulation.
Step 3	`Router(config-if)# dialer pool-member number`	Makes the interface a member of the dialing pool that the dialer interface will use; make sure the *number* arguments have the same value.
Step 4	`Router(config-if)# ppp authentication chap`	Specifies CHAP authentication.

Configuring Interfaces to Use a Backup Interface

To configure one or more interfaces to use a backup interface, use the following commands beginning in global configuration mode:

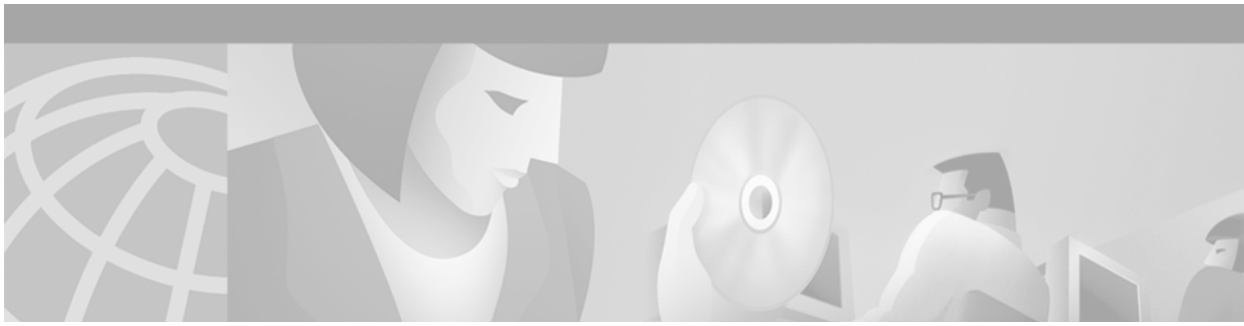
	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface to be backed up and begins interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered loopback0`	Specifies IP unnumbered loopback.

	Command	Purpose
Step 3	`Router(config-if)# backup interface dialer` *number*	Specifies the backup interface and begins interface configuration mode.
Step 4	`Router(config-if)# backup delay` *enable-delay disable-delay*	Specifies delay between the physical interface going down and the backup being enabled, and between the physical interface coming back up and the backup being disabled.

Configuration Example of Dialer Profile for ISDN BRI Backing Up Two Leased Lines

The following example shows the configuration of a site that backs up two leased lines using one BRI. Two dialer interfaces are defined. Each serial (leased line) interface is configured to use one of the dialer interfaces as a backup. Both of the dialer interfaces use dialer pool 1, which has physical interface BRI 0 as a member. Thus, physical interface BRI 0 can back up two different serial interfaces and can make calls to two different sites.

```
interface dialer0
 ip unnumbered loopback0
 encapsulation ppp
 dialer remote-name Remote0
 dialer pool 1
 dialer string 5551212
 dialer-group 1

interface dialer1
 ip unnumbered loopback0
 encapsulation ppp
 dialer remote-name Remote1
 dialer pool 1
 dialer string 5551234
 dialer-group 1

interface bri 0
 encapsulation PPP
 dialer pool-member 1
 ppp authentication chap

interface serial 0
 ip unnumbered loopback0
 backup interface dialer 0
 backup delay 5 10

interface serial 1
 ip unnumbered loopback0
 backup interface dialer1
 backup delay 5 10
```



Configuring Dial Backup Using Dialer Watch

This chapter describes how to configure dial backup using the Dialer Watch feature. It includes the following main sections:

- Dialer Watch Overview
- How to Configure Dialer Backup with Dialer Watch
- Configuration Examples for Dialer Watch

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the dial backup commands used to configure Dialer Watch, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Dialer Watch Overview

Dialer Watch is a backup feature that integrates dial backup with routing capabilities. Prior dial backup implementations used the following conditions to trigger backup:

- Interesting packets were defined at central and remote routers using dial-on-demand routing (DDR).
- Connection loss occurred on a primary interface using a back up interface with floating static routes.
- Traffic thresholds were exceeded using a dialer load threshold.

Prior backup implementations may not have supplied optimum performance on some networks, such as those using Frame Relay multipoint subinterfaces or Frame Relay connections that do not support end-to-end permanent virtual circuit (PVC) status updates.

Dialer Watch provides reliable connectivity without relying solely on defining interesting traffic to trigger outgoing calls at the central router. Dialer Watch uses the convergence times and characteristics of dynamic routing protocols. Integrating backup and routing features enables Dialer Watch to monitor every deleted route. By configuring a set of watched routes that define the primary interface, you are able to monitor and track the status of the primary interface as watched routes are added and deleted. Monitoring the watched routes is done in the following sequence:

1. Whenever a watched route is deleted, Dialer Watch checks whether there is at least one valid route for any of the defined watched IP addresses.
2. If no valid route exists, the primary line is considered down and unusable.

3. If a valid route exists for at least one of the defined IP addresses and if the route is pointing to an interface other than the backup interface configured for Dialer Watch, the primary link is considered up.
4. If the primary link goes down, Dialer Watch is immediately notified by the routing protocol and the secondary link is brought up.
5. Once the secondary link is up, at the expiration of each idle timeout, the primary link is rechecked.
6. If the primary link remains down, the idle timer is indefinitely reset.
7. If the primary link is up, the secondary backup link is disconnected. Additionally, you can set a disable timer to create a delay for the secondary link to disconnect, after the primary link is reestablished.

Dialer Watch provides the following advantages:

- Routing—Backup initialization is linked to the dynamic routing protocol, rather than a specific interface or static route entry. Therefore, both primary and backup interfaces can be any interface type, and can be used across multiple interfaces and multiple routers. Dialer Watch also relies on convergence, which is sometimes preferred over traditional DDR links.
- Routing protocol independent—Static routes or dynamic routing protocols, such as Interior Gateway Routing Protocol (IGRP), Enhanced IGRP (EIGRP) or Open Shortest Path First (OSPF) can be used.
- Nonpacket semantics—Dialer Watch does not exclusively rely on interesting packets to trigger dialing. The link is automatically brought up when the primary line goes down without postponing dialing.
- Dial backup reliability—DDR redial functionality is extended to dial indefinitely in the event that secondary backup lines are not initiated. Typically, DDR redial attempts are affected by enable-timeouts and wait-for-carrier time values. Intermittent media difficulties or flapping interfaces can cause problems for traditional DDR links. However, Dialer Watch automatically reestablishes the secondary backup line on ISDN, synchronous, and asynchronous serial links.

The following prerequisites apply to Dialer Watch:

- The router is dial backup capable, meaning the router has a data communications equipment (DCE), terminal adapter, or network termination 1 device attached that supports V.25*bis*.
- The router is configured for DDR. This configuration includes traditional commands such as **dialer map** and **dialer in-band** commands, and so on.
- Dialer Watch is only supported for IP at this time.

For information on how to configure traditional DDR for dial backup, see the other chapters in the "Dial Backup" part of this publication.

How to Configure Dialer Backup with Dialer Watch

To configure Dialer Watch, perform the following tasks. All tasks are required except the last one to set a disable timer.

- Determining the Primary and Secondary Interfaces (Required)
- Determining the Interface Addresses and Networks to Watch (Required)
- Configuring the Interface to Perform DDR Backup (Required)

- Creating a Dialer List (Required)
- Setting the Disable Timer on the Backup Interface (Optional)

Determining the Primary and Secondary Interfaces

Decide which interfaces on which routers will act as primary and secondary interfaces. Unlike traditional backup methods, you can define multiple interfaces on multiple routers instead of a singly defined interface on one router.

Determining the Interface Addresses and Networks to Watch

Determine which addresses and networks are to be monitored or watched. Typically, this will be the address of an interface on a remote router or a network advertised by a central or remote router.

Configuring the Interface to Perform DDR Backup

To initiate Dialer Watch, you must configure the interface to perform DDR and backup. Use traditional DDR configuration commands, such as dialer maps, for DDR capabilities. To enable Dialer Watch on the backup interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer watch-group group-number`	Enables Dialer Watch on the backup interface.

Creating a Dialer List

To define the IP addresses that you want watched, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer watch-list group-number ip ip-address address-mask`	Defines all IP addresses to be watched.

The **dialer watch-list** command is the means to detect if the primary interface is up or down. The primary interface is determined to be up when there is an available route with a valid metric to any of the addresses defined in this list, and it points to an interface other than the interface on which the **dialer watch-group** command is defined. The primary interface is determined to be down when there is no available route to any of the addresses defined in the **dialer watch-list** command.

Setting the Disable Timer on the Backup Interface

This task is optional. Under some conditions, you may want to implement a delay before the backup interface is dropped once the primary interface recovers. This delay can ensure stability, especially for flapping interfaces or interfaces experiencing frequent route changes.

Note The **dialer watch-disable** command used in Dialer Watch configurations was Replaced in Cisco IOS Release 12.3(11)T by the **dialer watch-list delay** command. When using the **dialer watch-list delay** command in software later than Cisco IOS Release 12.3(11)T, you can specify both a connect and disconnect timer for the disable timer. The disconnect time specifies that the disconnect timer is started when the secondary link is up and after the idle timeout period has expired, and only when software has determined that the primary route has come up

In Cisco IOS Software Releases Prior to 12.3(11)T

To apply a disable time, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer watch-disable seconds`	Applies a disable time to the interface.

In Cisco IOS Software Releases After 12.3(11)T

To apply a disable time, use the following command in global configuration mode:

Command	Purpose
`Router(config-if)# dialer watch-list group-number delay {connect connect-time | disconnect disconnect-time}`	Configures a disable time. • *group-number*—Group number assigned to the list. Valid group numbers are from 1 to 255. • **delay**—Specifies that the router will delay dialing the secondary link when the primary link becomes unavailable. • **connect** *connect-time*—Time, in seconds, after which the router rechecks for availability of the primary link. If the primary link is still unavailable, the secondary link is then dialed. Valid times range from 1 to 2147483 seconds. • **disconnect** *disconnect-time*—Time, in seconds, that specifies when to disconnect. Disconnect occurs when the secondary link is up and after the idle timeout period has expired, and only when software has determined that the primary route has come up. Valid times range from 1 to 2147483 seconds.

Configuration Examples for Dialer Watch

The **dialer watch-disable** command used in Dialer Watch configurations was replaced in Cisco IOS Release 12.3(11)T by the **dialer watch-list delay** command. The following sections provide examples of how to configure Dialer Watch in software before and after the **dialer watch-disable** command was replaced.

- Dialer Watch Configuration Example Prior to Cisco IOS Release 12.3(11)T, page 463
- Dialer Watch Configuration Example After Cisco IOS Release 12.3(11)T, page 467

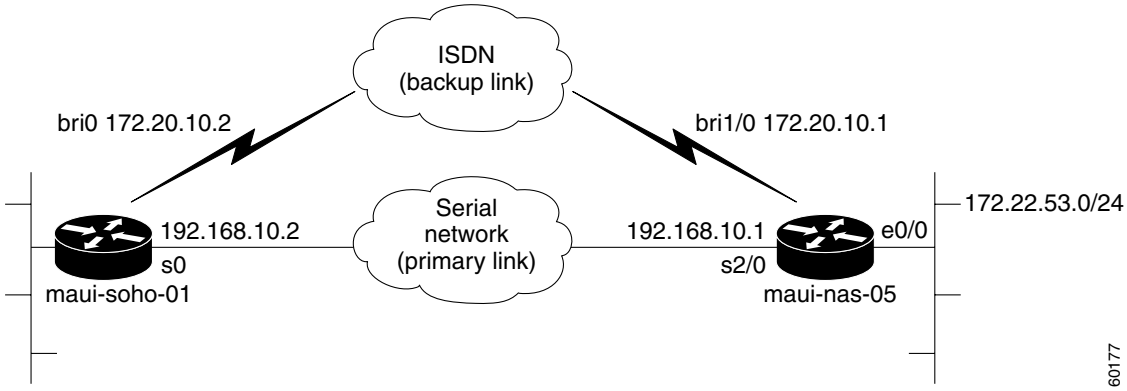
Dialer Watch Configuration Example Prior to Cisco IOS Release 12.3(11)T

In the following example, an ISDN BRI line is used to back up a serial leased line connection by configuring the Dialer Watch feature on a router named maui-soho-01. The Dialer Watch feature enables the router to monitor the existence of a specified route. If that route is not present, the backup interface is activated. Unlike other backup methods, the Dialer Watch feature does not require interesting traffic to activate the backup interface. The configuration shown in Figure 61 uses legacy dial-on-demand routing (DDR) and the Open Shortest Path First (OSPF) routing protocol. Dialer profiles can be used in place of DDR. Once the backup connection is activated, you must ensure that the routing table is updated to use the new backup route. Additional information about the Dialer Watch feature is available at the following website:

http://www.cisco.com/warp/public/129/bri-backup-map-watch.html

For additional information on configuring legacy DDR, dialer profiles, PPP, and traditional dial backup features, see the relevant chapters in this publication.

Figure 61 Dialer Watch for Frame Relay Interfaces



Note The following example uses commands supported in Cisco IOS software prior to Release 12.3(11)T. See the updated example for configuring Dialer Watch after Cisco IOS Release 12.3(11)T that follows this example.

Configuration for maui-soho-01

```
maui-soho-01# show running-config

Building configuration...

Current configuration : 1546 bytes
!
version 12.1
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname maui-soho-01
!
logging rate-limit console 10 except errors
aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHEN none
aaa authentication ppp default local
```

```
!This is basic AAA configuration for PPP calls.
enable secret 5 <deleted>
!
username maui-nas-05 password 0 cisco
!Username for remote router (maui-nas-05) and shared secret.
!Shared secret(used for CHAP authentication) must be the same on both sides.
ip subnet-zero
no ip finger
!
isdn switch-type basic-ni
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.0
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0
!Primary link.
 ip address 192.168.10.2 255.255.255.252
 encapsulation ppp
 ppp authentication chap
!
interface BRI0
 ip address 172.20.10.2 255.255.255.0
!IP address for the BRI interface (backup link).
 encapsulation ppp
 dialer idle-timeout 30
!Idle timeout(in seconds)for this backup link.
!Dialer watch checks the status of the primary link every time the
!idle-timeout expires.
 dialer watch-disable 15
!Delays disconnecting the backup interface for 15 seconds after the
!primary interface is found to be up.
 dialer map ip 172.20.10.1 name maui-nas-05 broadcast 5550111
!Dialer map for the BRI interface of the remote router.
 dialer map ip 172.22.53.0 name maui-nas-05 broadcast 5550111
!Map statement for the route/network being watched by the
!dialer watch-list command.
!This address must exactly match the network configured with the
!dialer watch-list command.
!When the watched route disappears, this dials the specified phone number.
 dialer watch-group 8
!Enable Dialer Watch on this backup interface.
!Watch the route specified with dialer watch-list 8.
 dialer-group 1
!Apply interesting traffic defined in dialer-list 1.
 isdn switch-type basic-ni
 isdn spid1 51255522220101 5550112
 isdn spid2 51255522230101 5550112
 ppp authentication chap
!Use chap authentication.
!
router ospf 5
 log-adjacency-changes
 network 172.16.1.0 0.0.0.255 area 0
 network 172.17.1.0 0.0.0.255 area 0
 network 172.20.10.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.3 area 0
!
ip classless
no ip http server
!
```

```
dialer watch-list 8 ip 172.22.53.0 255.255.255.0
!This defines the route(s) to be watched.
!This exact route(including subnet mask) must exist in the routing table.
!Use the dialer watch-group 8 command to apply this list to the backup interface.
access-list 101 remark Define Interesting Traffic
access-list 101 deny   ospf any any
!Mark OSPF as uninteresting.
!This will prevent OSPF hellos from keeping the link up.
Access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!Interesting traffic is defined by access-list 101.
!This is applied to BRI0 using dialer-group 1.
!
line con 0
 login authentication NO_AUTHEN
 transport input none
line vty 0 4
!
end
```

Configuration for maui-nas-05

```
maui-nas-05# show running-config

Building configuration...

Current configuration:
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname maui-nas-05
!
aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHEN none
aaa authentication ppp default local
! -- This is basic AAA configuration for PPP calls.
Enable secret 5 <deleted>
!
username maui-soho-01 password 0 cisco
!Username for remote router (maui-soho-01) and shared secret.
!Shared secret(used for CHAP authentication) must be the same on both sides.
!
ip subnet-zero
!
isdn switch-type basic-ni
!
interface Loopback0
 ip address 172.22.1.1 255.255.255.0
!
interface Ethernet0/0
 ip address 172.22.53.105 255.255.255.0
!
interface Ethernet0/1
 no ip address
 shutdown
!
interface BRI1/0
!Backup link.
 ip address 172.20.10.1 255.255.255.0
 encapsulation ppp
```

```
 dialer map ip 172.20.10.2 name maui-soho-01 broadcast
!Dialer map with IP address and authenticated username for remote destination.
!The name should match the authentication username provided by the remote side.
!The dialer map statement is used even though this router is not dialing out.
 Dialer-group 1
!Apply interesting traffic defined in dialer-list 1.
 isdn switch-type basic-ni
 isdn spid1 51255501110101 5550111
 isdn spid2 51255501120101 5550112
 ppp authentication chap
!
.
.
.
!
interface Serial2/0
 ip address 192.168.10.1 255.255.255.252
 encapsulation ppp
 clockrate 64000
 ppp authentication chap
!
.
.
.
!
router ospf 5
 network 172.20.10.0 0.0.0.255 area 0
 network 172.22.1.0 0.0.0.255 area 0
 network 172.22.53.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.3 area 0
 default-information originate
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
no ip http server
!
dialer-list 1 protocol ip permit
!This defines all IP traffic as interesting.
!
line con 0
 login authentication NO_AUTHEN
 transport input none
line 97 102
line aux 0
line vty 0 4
!
end
```

Dialer Watch Configuration Example After Cisco IOS Release 12.3(11)T

The following example shows how to configure Dialer Watch using the **dialer watch-list delay** command that replaced the **dialer watch-disable** command.

Configuration for maui-soho-01

```
maui-soho-01# show running-config
Building configuration...

Current configuration : 1546 bytes
!
version 12.4
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname maui-soho-01
!
logging rate-limit console 10 except errors
aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHEN none
aaa authentication ppp default local
!This is basic AAA configuration for PPP calls.
enable secret 5 <deleted>
!
username maui-nas-05 password 0 cisco
!Username for remote router (maui-nas-05) and shared secret.
!Shared secret(used for CHAP authentication) must be the same on both sides.
ip subnet-zero
no ip finger
!
isdn switch-type basic-ni
!
interface Loopback0
 ip address 172.17.1.1 255.255.255.0
!
interface Ethernet0
 ip address 172.16.1.1 255.255.255.0
!
interface Serial0
!Primary link.
 ip address 192.168.10.2 255.255.255.252
 encapsulation ppp
 ppp authentication chap
!
interface BRI0
 ip address 172.20.10.2 255.255.255.0
!IP address for the BRI interface (backup link).
 encapsulation ppp
 dialer idle-timeout 30
!Idle timeout(in seconds)for this backup link.
!Dialer watch checks the status of the primary link every time the
!idle-timeout expires.
 dialer map ip 172.20.10.1 name maui-nas-05 broadcast 5550111
!Dialer map for the BRI interface of the remote router.
 dialer map ip 172.22.53.0 name maui-nas-05 broadcast 5550111
!Map statement for the route/network being watched by the
!dialer watch-list command.
!This address must exactly match the network configured with the
!dialer watch-list command.
```

```
!When the watched route disappears, this dials the specified phone number.
 dialer watch-group 8
!Enable Dialer Watch on this backup interface.
!Watch the route specified with dialer watch-list 8.
 dialer-group 1
!Apply interesting traffic defined in dialer-list 1.
isdn switch-type basic-ni
 isdn spid1 51255522220101 5552222
 isdn spid2 51255522230101 5552223
 ppp authentication chap
!Use chap authentication.
dialer watch-list 8 delay disconnect 15
!Delays disconnecting the backup interface for 15 seconds after the
!primary interface is found to be up.
!
router ospf 5
 log-adjacency-changes
 network 172.16.1.0 0.0.0.255 area 0
 network 172.17.1.0 0.0.0.255 area 0
 network 172.20.10.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.3 area 0
!
ip classless
no ip http server
!
dialer watch-list 8 ip 172.22.53.0 255.255.255.0
!This defines the route(s) to be watched.
!This exact route(including subnet mask) must exist in the routing table.
!Use the dialer watch-group 8 command to apply this list to the backup interface.
access-list 101 remark Define Interesting Traffic
access-list 101 deny   ospf any any
!Mark OSPF as uninteresting.
!This will prevent OSPF hellos from keeping the link up.
Access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
!Interesting traffic is defined by access-list 101.
!This is applied to BRI0 using dialer-group 1.
!
line con 0
 login authentication NO_AUTHEN
 transport input none
line vty 0 4
!
end
```

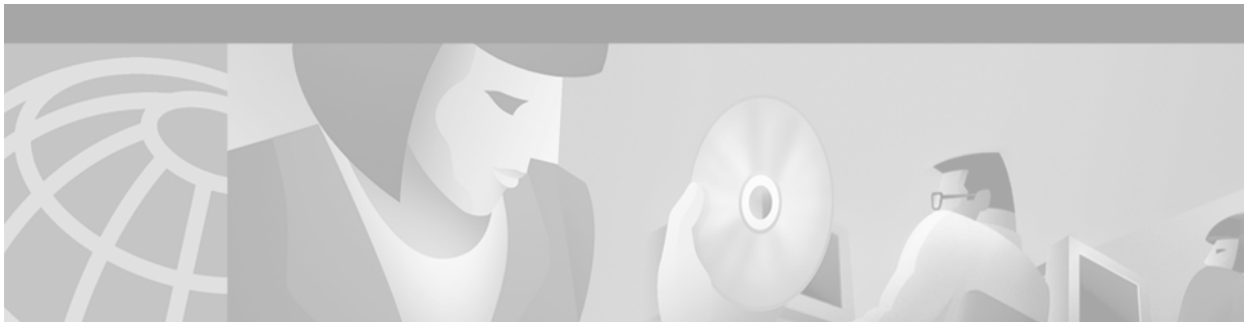
Configuration for maui-nas-05

```
maui-nas-05# show running-config
Building configuration...

Current configuration:
!
version 12.4
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname maui-nas-05
!
aaa new-model
aaa authentication login default local
aaa authentication login NO_AUTHEN none
aaa authentication ppp default local
```

```
! -- This is basic AAA configuration for PPP calls.
Enable secret 5 <deleted>
!
username maui-soho-01 password 0 cisco
!Username for remote router (maui-soho-01) and shared secret.
!Shared secret(used for CHAP authentication) must be the same on both sides.
!
ip subnet-zero
!
isdn switch-type basic-ni
!
interface Loopback0
 ip address 172.22.1.1 255.255.255.0
!
interface Ethernet0/0
 ip address 172.22.53.105 255.255.255.0
!
interface Ethernet0/1
 no ip address
 shutdown
!
interface BRI1/0
!Backup link.
 ip address 172.20.10.1 255.255.255.0
 encapsulation ppp
 dialer map ip 172.20.10.2 name maui-soho-01 broadcast
!Dialer map with IP address and authenticated username for remote destination.
!The name should match the authentication username provided by the remote side.
!The dialer map statement is used even though this router is not dialing out.
 Dialer-group 1
!Apply interesting traffic defined in dialer-list 1.
 isdn switch-type basic-ni
 isdn spid1 51255501110101 5550111
 isdn spid2 51255501120101 5550112
 ppp authentication chap
!
! <<-- irrelevant output removed
!
interface Serial2/0
 ip address 192.168.10.1 255.255.255.252
 encapsulation ppp
 clockrate 64000
 ppp authentication chap
!
! <<-- irrelevant output removed
!
router ospf 5
 network 172.20.10.0 0.0.0.255 area 0
 network 172.22.1.0 0.0.0.255 area 0
 network 172.22.53.0 0.0.0.255 area 0
 network 192.168.10.0 0.0.0.3 area 0
 default-information originate
!
ip classless
ip route 0.0.0.0 0.0.0.0 Ethernet0/0
no ip http server
!
dialer-list 1 protocol ip permit
!This defines all IP traffic as interesting.
!
line con 0
 login authentication NO_AUTHEN
 transport input none
line 97 102
```

```
line aux 0
line vty 0 4
!
end
```

Dial-Related Addressing Services



Configuring Cisco Easy IP

This chapter describes how to configure the Cisco Easy IP feature. It includes the following main sections:

- Cisco Easy IP Overview
- How to Configure Cisco Easy IP
- Configuration Examples for Cisco Easy IP

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the Cisco Easy IP commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Cisco Easy IP Overview

Cisco Easy IP enables transparent and dynamic IP address allocation for hosts in remote environments using the following functionality:

- Cisco Dynamic Host Configuration Protocol (DHCP) server
- Port Address Translation (PAT), a subset of Network Address Translation (NAT)
- Dynamic PPP/IP Control Protocol (PPP/IPCP) WAN interface IP address negotiation

With the Cisco IOS Easy IP, a Cisco router automatically assigns local IP addresses to remote hosts (such as small office, home office or SOHO routers) using DHCP with the Cisco IOS DHCP server, automatically negotiates its own registered IP address from a central server via PPP/IPCP, and uses PAT functionality to enable all SOHO hosts to access the Internet using a single registered IP address. Because Cisco IOS Easy IP uses existing port-level multiplexed NAT functionality within Cisco IOS software, IP addresses on the remote LAN are invisible to the Internet, making the remote LAN more secure.

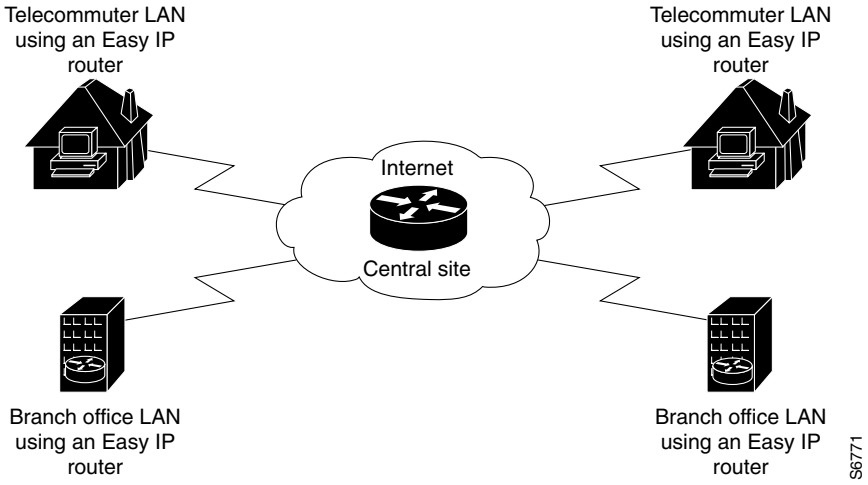
Cisco Easy IP provides the following benefits:

- Minimizes Internet access costs for remote offices
- Minimizes configuration requirements on remote access routers
- Enables transparent and dynamic IP address allocation for hosts in remote environments
- Improves network security capabilities at each remote site

- Conserves registered IP addresses
- Maximizes IP address manageability

Figure 62 shows a typical scenario for using the Cisco Easy IP feature.

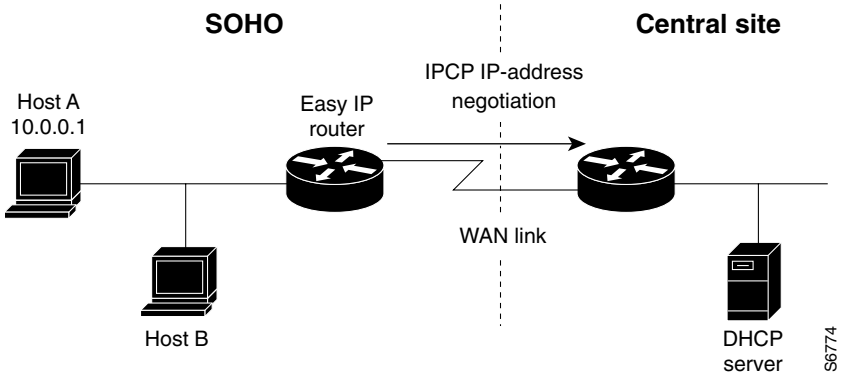
Figure 62 Telecommuter and Branch Office LANs Using Cisco Easy IP



Steps 1 through 4 show how Cisco Easy IP works:

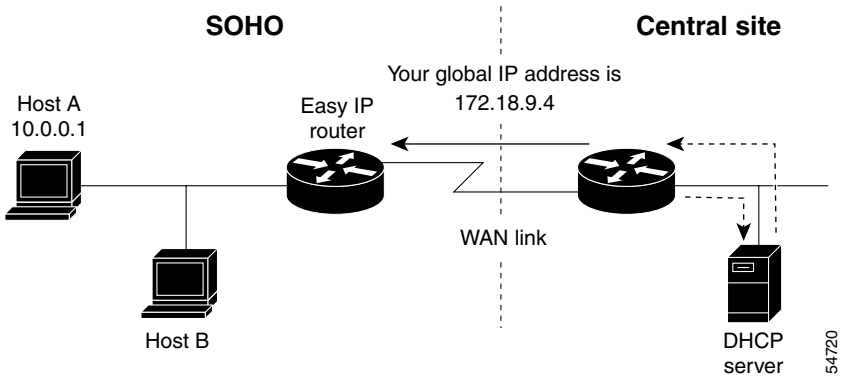
Step 1 When a SOHO host generates "interesting" traffic (as defined by Access Control Lists) for dialup (first time only), the Easy IP router requests a single registered IP address from the access server at the central site via PPP/IPCP. (See Figure 63.)

Figure 63 Cisco Easy IP Router Requests a Dynamic Global IP Address



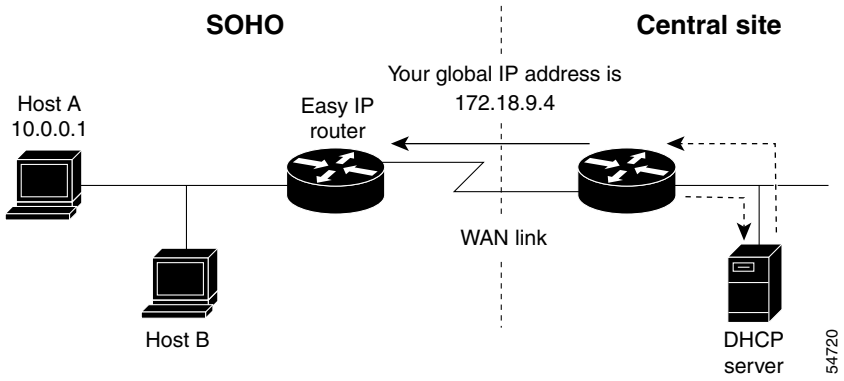
Step 2 The central site router replies with a dynamic global address from a local DHCP IP address pool. (See Figure 64.)

Figure 64 Dynamic Global IP Address Delivered to the Cisco Easy IP Router



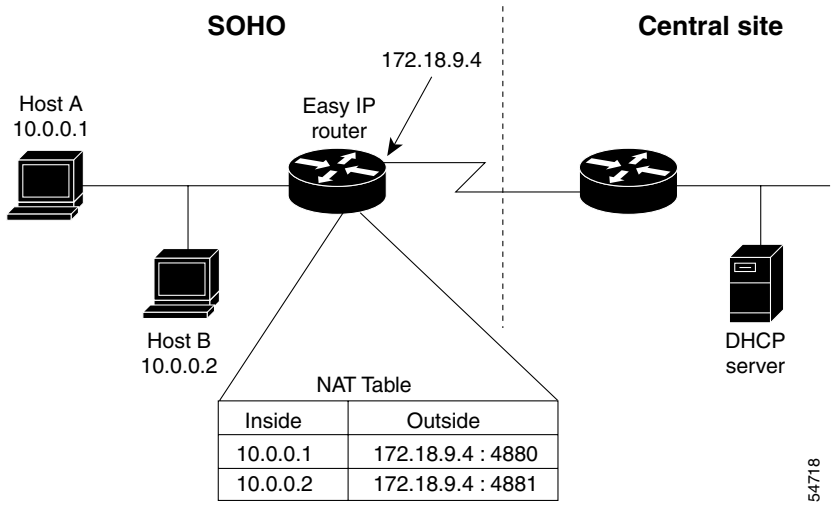
Step 3 The Cisco Easy IP router uses port-level NAT functionality to automatically create a translation that associates the registered IP address of the WAN interface with the private IP address of the client. (See Figure 65.)

Figure 65 Port-Level NAT Functionality Used for IP Address Translation



Step 4 The remote hosts contain multiple static IP addresses while the Cisco Easy IP router obtains a single registered IP address using PPP/IPCP. The Cisco Easy IP router then creates port-level multiplexed NAT translations between these addresses so that each remote host address (inside private address) is translated to a single external address assigned to the Cisco Easy IP router. This many-to-one address translation is also called port-level multiplexing or PAT. Note that the NAT port-level multiplexing function can be used to conserve global addresses by allowing the remote routers to use one global address for many local addresses. (See Figure 66.)

Figure 66 Multiple Private Internal IP Addresses Bound to a Single Global IP Address



How to Configure Cisco Easy IP

Before using Cisco Easy IP, perform the following tasks:

- Configure the ISDN switch type and service provider identifier (SPID), if using ISDN.
- Configure the static route from LAN to WAN interface.
- Configure the Cisco IOS DHCP server.

For information about configuring ISDN switch types, see the chapter "Setting Up ISDN Basic Rate Service" earlier in this publication. For information about configuring static routes, refer to the chapter "Configuring IP Services" in the *Cisco IOS IP Configuration Guide*.

The Cisco IOS DHCP server supports both DHCP and BOOTP clients and supports finite and infinite address lease periods. DHCP address binding information is stored on a remote host via remote copy protocol (RCP), FTP, or TFTP. Refer to the *Cisco IOS IP Configuration Guide* for DHCP configuration instructions.

In its most simple configuration, a Cisco Easy IP router or access server will have a single LAN interface and a single WAN interface. Based on this model, to use Cisco Easy IP you must perform the tasks in the following sections:

- Defining the NAT Pool (Required)
- Configuring the LAN Interface (Required)
- Defining NAT for the LAN Interface (Required)
- Configuring the WAN Interface (Required)
- Enabling PPP/IPCP Negotiation (Required)
- Defining NAT for the Dialer Interface (Required)
- Configuring the Dialer Interface (Required)

For configuration examples, see the section "Configuration Examples for Cisco Easy IP" at the end of this chapter.

Defining the NAT Pool

The first step in enabling Cisco Easy IP is to create a pool of internal IP addresses to be translated. To define the NAT pool, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number permit source [source-wildcard]`	Defines a standard access list permitting those addresses that are to be translated.
Step 2	`Router(config)# ip nat inside source list access-list-number interface dialer-name overload`	Establishes dynamic source translation, identifying the access list defined in the prior step.

For information about creating access lists, refer to the chapter "Configuring IP Services" in the *Cisco IOS IP Configuration Guide*.

Configuring the LAN Interface

To configure the LAN interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Selects a specific LAN interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address address mask`	Defines the IP address and subnet mask for this interface.

For information about assigning IP addresses and subnet masks to network interfaces, refer to the chapter "Configuring IP Services" in the *Cisco IOS IP Configuration Guide*.

Defining NAT for the LAN Interface

To ensure that the LAN interface is connected to the inside network (and therefore subject to NAT), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip nat inside`	Defines the interface as internal for NAT.

Configuring the WAN Interface

To configure the WAN interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Selects the WAN interface and begins interface configuration mode.
Step 2	`Router(config-if)# no ip address`	Removes any associated IP address from this interface.

	Command	Purpose
Step 3	`Router(config-if)# encapsulation ppp`	Selects PPP as the encapsulation method for this interface.
Step 4	`Router(config-if)# dialer pool-member` *number*	Binds the WAN interface to the dialer interface.

Enabling PPP/IPCP Negotiation

To enable PPP/IPCP negotiation on the dialer interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *dialer-name*	Selects the dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address negotiated`	Enables PPP/IPCP negotiation for this interface.

Defining NAT for the Dialer Interface

To define that the dialer interface is connected to the outside network, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *dialer-name*	Selects the dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip nat outside`	Defines the interface as external for network address translation.

Configuring the Dialer Interface

To configure the dialer interface information, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *dialer-name*	Selects the dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# dialer wait-for-carrier-time` *seconds*	Specifies for a dialer interface the length of time the interface waits for a carrier before timing out.
Step 3	`Router(config-if)# dialer hold-queue` *packets*	Creates a dialer hold queue and specifies the number of packets to be held in it.
Step 4	`Router(config-if)# dialer remote-name` *username*	Specifies the remote router Challenge Handshake Authentication Protocol (CHAP) authentication name.

	Command	Purpose
Step 5	`Router(config-if)# dialer idle-timeout seconds`	Specifies the amount of idle time that can pass before calls to the central access server are disconnected. See the next section "Timeout Considerations," for more details on this setting.
Step 6	`Router(config-if)# dialer string dialer-string`	Specifies the telephone number required to reach the central access server.
Step 7	`Router(config-if)# dialer pool number`	Specifies the dialing pool to use.
Step 8	`Router(config-if)# dialer-group group-number`	Assigns the dialer interface to a dialer group.

Timeout Considerations

Dynamic NAT translations time out automatically after a predefined default period. Although configurable, with the port-level NAT functionality in Cisco Easy IP, Domain Name System (DNS) User Datagram Protocol (UDP) translations time out after 5 minutes, while DNS translations time out after 1 minute by default. TCP translations time out after 24 hours by default, unless a TCP Reset (RST) or TCP Finish (FIN) is seen in the TCP stream, in which case the translation times out after 1 minute.

If the Cisco IOS Easy IP router exceeds the dialer idle-timeout period, it is expected that all active TCP sessions were previously closed via an RST or FIN. NAT times out all TCP translations before the Cisco Easy IP router exceeds the dialer idle-timeout period. The router then renegotiates another registered IP address the next time the WAN link is brought up, thereby creating new dynamic NAT translations that bind the IP addresses of the LAN host to the newly negotiated IP address.

Configuration Examples for Cisco Easy IP

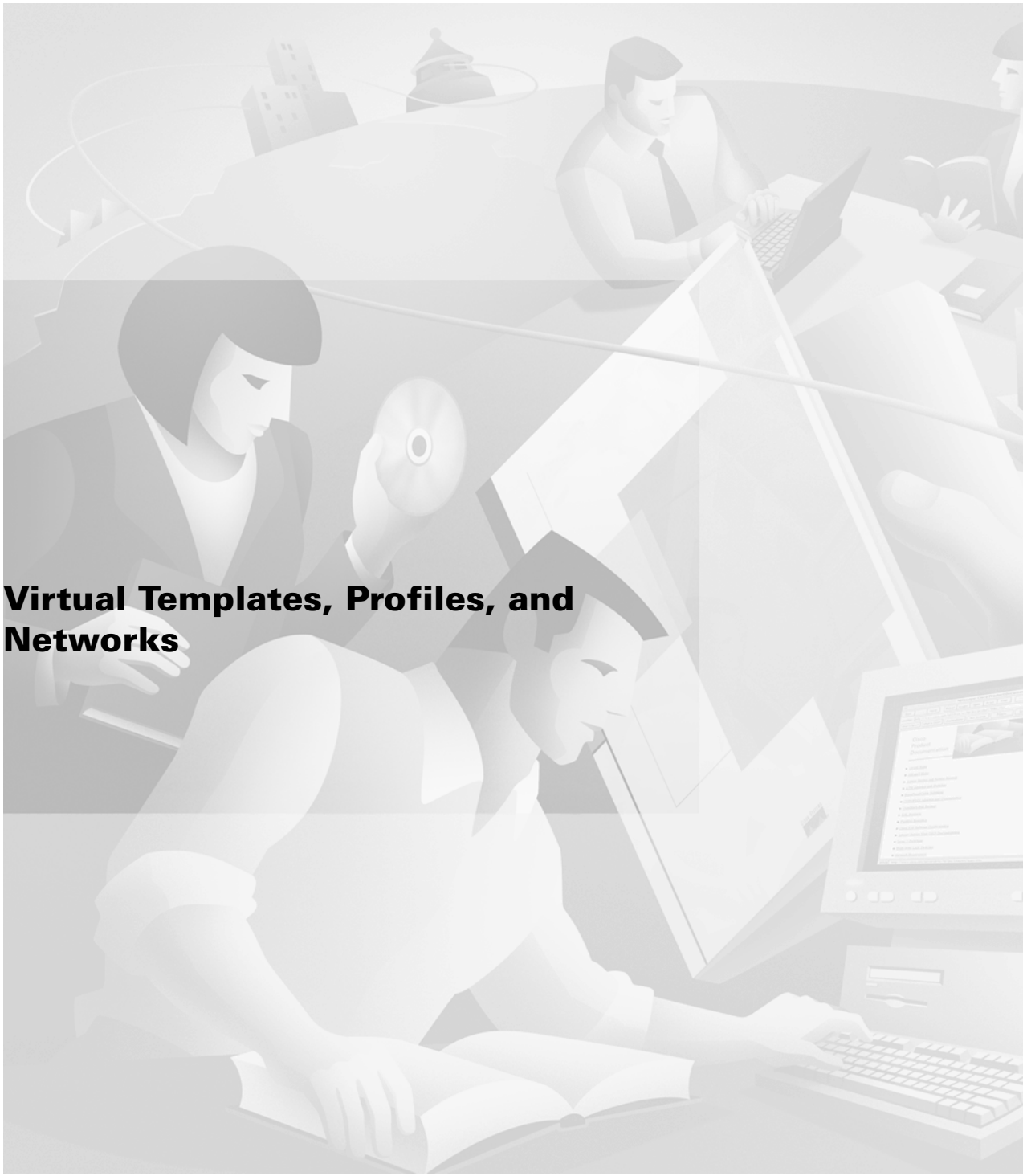
The following example shows how to configure BRI interface 0 (shown as interface bri0) to obtain its IP address via PPP/IPCP address negotiation:

```
! The following command defines the NAT pool.
ip nat inside source list 101 interface dialer1 overload
!
! The following commands define the ISDN switch type.
isdn switch type vn3
isdn tei-negotiation first-call
!
! The following commands define the LAN address and subnet mask.
interface ethernet0
 ip address 10.0.0.4 255.0.0.0

! The following command defines ethernet0 as internal for NAT.
 ip nat inside
!
! The following commands binds the physical interface to the dialer1 interface.
interface bri0
 no ip address
 encapsulation ppp
 dialer pool-member 1
!
interface dialer1
!
! The following command enables PPP/IPCP negotiation for this interface.
 ip address negotiated
 encapsulation ppp
```

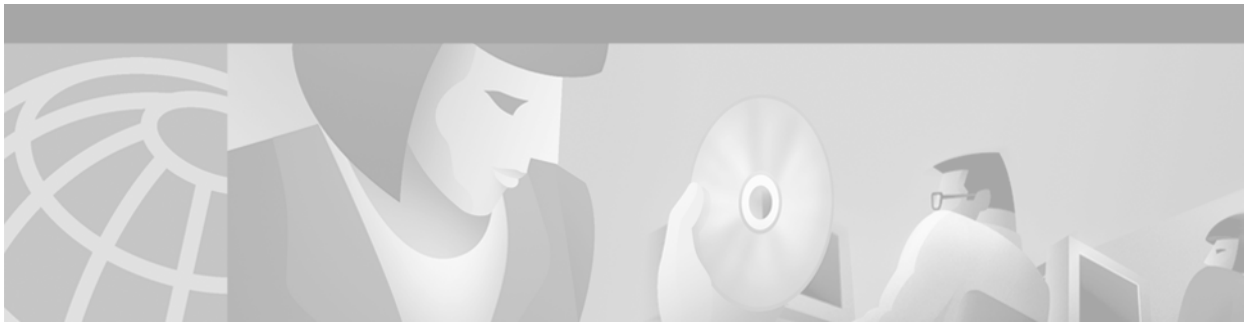
```
!
! The following command defines interface dialer1 as external for NAT.
 ip nat outside
 dialer remote-name dallas
 dialer idle-timeout 180
!
! The following command defines the dialer string for the central access server.
 dialer string 4159991234
 dialer pool 1
 dialer-group 1
!
! The following commands define the static route to the WAN interface.
ip route 0.0.0.0 0.0.0.0 dialer1
access-list 101 permit ip 10.0.0.0 0.255.255.255 any
dialer-list 1 protocol ip list 101
```

The following example shows how to configure an asynchronous interface (interface async1) to obtain its IP address via PPP/IPCP address negotiation:

```
! This command defines the NAT pool.
ip nat inside source list 101 interface dialer 1 overload
!
! The following commands define the LAN IP address and subnet mask.
interface ethernet0
 ip address 10.0.0.4 255.0.0.0
!
! The following command defines ethernet0 as internal for NAT.
 ip nat inside
!
! The following commands bind the physical dialer1 interface.
interface async1
 no ip address
 encapsulation ppp
 async mode dedicated
 dialer pool-member 1
!
interface dialer1
!
! The following command enables PPP/IPCP negotiation for this interface.
 ip address negotiated
 encapsulation ppp
!
! The following command defines interface dialer1 as external for NAT.
 ip nat outside
 dialer wait-for-carrier-time 30
 dialer hold-queue 10
 dialer remote-name dallas
 dialer idle-timeout 180
!
! The following command defines the dialer string for the central access server.
 dialer string 4159991234
 dialer pool 1
 dialer-group 1
!
! The following commands define the static route to the WAN interface.
ip route 0.0.0.0 0.0.0.0 dialer1
access-list 101 permit ip 10.0.0.0 0.255.255.255 any
dialer-list 1 protocol ip list 101
```



Virtual Templates, Profiles, and Networks



Configuring Virtual Template Interfaces

This chapter describes how to configure virtual template interfaces. It includes the following main sections:

The following template and virtual interface limitations apply:

- Although a system can generally support many virtual template interfaces, one template for each virtual access application is a more realistic limit.
- When in use, each virtual access interface cloned from a template requires the same amount of memory as a serial interface. Limits to the number of virtual access interfaces that can be configured are determined by the platform.
- Virtual access interfaces are not directly configurable by users, except by configuring a virtual template interface or including the configuration information of the user (through virtual profiles or per-user configuration) on an authentication, authorization, and accounting (AAA) server. However, information about an in-use virtual access interface can be displayed, and the virtual access interface can be cleared.
- Virtual interface templates provide no *direct* value to users; they must be applied to or associated with a virtual access feature using a command with the **virtual-template** keyword.

  For example, the **interface virtual-template** command creates the virtual template interface and the **multilink virtual-template** command applies the virtual template to a multilink stack group. The **virtual-profile virtual-template** command specifies that a virtual template interface will be used as a source of configuration information for virtual profiles.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the virtual template interface commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Virtual Template Interface Service Overview

The Virtual Template Interface Service feature provides a generic service that can be used to apply predefined interface configurations (virtual template interfaces) in creating and freeing virtual access interfaces dynamically, as needed.

Virtual template interfaces can be configured independently of any physical interface and applied dynamically, as needed, to create virtual access interfaces. When a user dials in, a predefined configuration template is used to configure a virtual access interface; when the user is done, the virtual access interface goes down and the resources are freed for other dial-in uses.

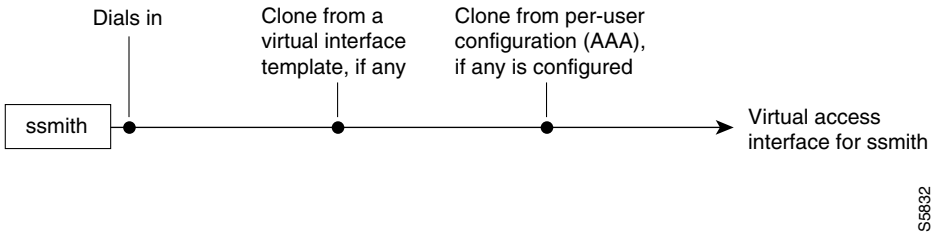
A virtual template interface is a logical entity—a configuration for a serial interface but not tied to a physical interface—that can be applied dynamically as needed. Virtual access interfaces are virtual interfaces that are created, configured dynamically (for example, by *cloning* a virtual template interface), used, and then freed when no longer needed.

Virtual template interfaces are one possible source of configuration information for a virtual access interface.

Each virtual access interface can clone from only one template. But some applications can take configuration information from multiple sources; for example, virtual profiles can take configuration information from a virtual template interface, or from interface-specific configuration information stored from a user on a AAA server, or from network protocol configuration from a user stored on a AAA server, or all three. The result of using template and AAA configuration sources is a virtual access interface uniquely configured for a specific dial-in user.

Figure 67 illustrates that a router can create a virtual access interface by first using the information from a virtual template interface (if any is defined for the application) and then using the information in a per-user configuration (if AAA is configured on the router and virtual profiles or per-user configuration or both are defined for the specific user).

Figure 67 Possible Configuration Sources for Virtual Access Interfaces



The virtual template interface service is intended primarily for customers with large numbers of dial-in users and provides the following benefits:

- For easier maintenance, allows customized configurations to be predefined and then applied dynamically when the specific need arises.
- For scalability, allows interface configuration to be separated from physical interfaces. Virtual interfaces can share characteristics, no matter what specific type of interface the user called on.
- For consistency and configuration ease, allows the same predefined template to be used for all users dialing in for a specific application.
- For efficient router operation, frees the virtual access interface memory for another dial-in use when the call from the user ends.

Features that Apply Virtual Template Interfaces

The following features apply virtual template interfaces to create virtual access interfaces dynamically:

- Virtual profiles
- Virtual Private Dialup Networks (VPDN)
- Multilink PPP (MLP)
- Multichassis Multilink PPP (MMP)
- Virtual templates for protocol translation
- PPP over ATM

Virtual templates are supported on all platforms that support these features.

To create and configure a virtual template interface, compete the tasks in this chapter. To apply a virtual template interface, refer to the specific feature that applies the virtual template interface.

All prerequisites depend on the feature that is applying a virtual template interface to create a virtual access interface. Virtual template interfaces themselves have no other prerequisites.

The order in which you create virtual template interfaces and virtual profiles and configure the features that use the templates and profiles is not important. They must exist, however, before someone calling in can use them.

Selective Virtual Access Interface Creation

Optionally, you can configure a router to automatically determine whether to create a virtual access interface for each inbound connection. In particular, a call that is received on a physical asynchronous interface that uses a AAA per-user configuration can now be processed without a virtual access interface being created by a router that is also configured for virtual profiles.

The following three criteria determine whether a virtual access interface is created:

- Is there a virtual profile AAA configuration?
- Is there a AAA per-user configuration?
- Does the link interface support direct per-user AAA?

A virtual access interface *will* be created in the following scenarios:

- If there *is* a virtual profile AAA configuration.
- If there *is not* a virtual profile AAA configuration, but there *is* a AAA per-user configuration *and* the link interface *does not* support direct per-user AAA (such as ISDN).

A virtual access interface *will not* be created in the following scenarios:

- If there is *neither* a virtual profile AAA configuration *nor* a AAA per-user configuration.
- If there is *not* a virtual profile AAA configuration, but there *is* a AAA per-user configuration and the link interface does support direct per-user AAA (such as asynchronous).

How to Configure a Virtual Template Interface

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered ethernet 0`	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template Interface.
Step 4	`Router(config-if)# virtual-profile if-needed`	(Optional) Creates virtual-access interfaces only if the inbound connection requires one.

Note Configuring the **ip address** command within a virtual template is not recommended. Configuring a specific IP address in a virtual template can result in the establishment of erroneous routes and the loss of IP packets.

Optionally, other PPP configuration commands can be added to the virtual template configuration. For example, you can add the **ppp authentication chap** command.

All configuration commands that apply to serial interfaces can also be applied to virtual template interfaces, except **shutdown** and **dialer** commands.

For virtual template interface examples, see the "Configuration Examples for Virtual Template Interface" section later in this chapter.

Monitoring and Maintaining a Virtual Access Interface

When a virtual template interface or a configuration from a user on a AAA server or both are applied dynamically, a virtual access interface is created. Although a virtual access interface cannot be created and configured directly, it can be displayed and cleared.

To display or clear a specific virtual access interface, use the following commands in EXEC mode:

Command	Purpose
`Router> show interfaces virtual-access number`	Displays the configuration of the virtual access interface.
`Router> clear interface virtual-access number`	Tears down the virtual access interface and frees the memory for other dial-in uses.

Configuration Examples for Virtual Template Interface

The following sections provide virtual template interface configuration examples:

- Basic PPP Virtual Template Interface
- Virtual Template Interface

Basic PPP Virtual Template Interface

The following example enables virtual profiles (configured only by virtual template) on straightforward PPP (no MLP), and configures a virtual template interface that can be cloned on a virtual access interface for dial-in users:

```
virtual-profile virtual-template 1

interface virtual-template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp authentication chap
```

Virtual Template Interface

The following two examples configure a virtual template interface and then display the configuration of a virtual access interface when the template interface has been applied.

This example uses a named Internet Protocol Exchange (IPX) access list:

```
Router(config)# interface virtual-template 1
 ip unnumbered Ethernet0
 ipx ppp-client Loopback2
 no cdp enable
 ppp authentication chap
```

This example displays the configuration of the active virtual access interface that was configured by virtual-template 1, defined in the preceding example:

```
Router# show interfaces virtual-access 1 configuration

Virtual-Access1 is a L2F link interface
interface Virtual-Access1 configuration...
ip unnumbered Ethernet0
ipx ppp-client Loopback2
no cdp enable
ppp authentication chap
```

Selective Virtual Access Interface

The following example shows how to create a virtual access interface for incoming calls that require a virtual access interface:

```
aaa new-model
aaa authentication ppp default local radius tacacs
aaa authorization network default local radius tacacs

virtual-profile if-needed
virtual-profile virtual-template 1
virtual-profile aaa
!
interface Virtual-Template1
```

```
 ip unnumbered Ethernet 0
 no ip directed-broadcast
 no keepalive
 ppp authentication chap
 ppp multilink
```

RADIUS Per-User and Virtual Profiles

The following examples show RADIUS user profiles that could be used for selective virtual access interface creation.

This example shows AAA per-user configuration for a RADIUS user profile:

```
RADIUS user profile:
        foo     Password = "test"
                 User-Service-Type = Framed-User,
                 Framed-Protocol = PPP,
                cisco-avpair = "ip:inacl#1=deny 10.10.10.10 0.0.0.0",
                cisco-avpair = "ip:inacl#1=permit any"
```

This example shows a virtual profile AAA configuration for a RADIUS user profile:

```
RADIUS user profile:
       foo  Password = "test"
            User-Service-Type = Framed-User,
            Framed-Protocol = PPP,
          cisco-avpair = "lcp:interface-config=keepalive 30\nppp max-bad-auth 4"
```

TACACS+ Per-User and Virtual Profiles

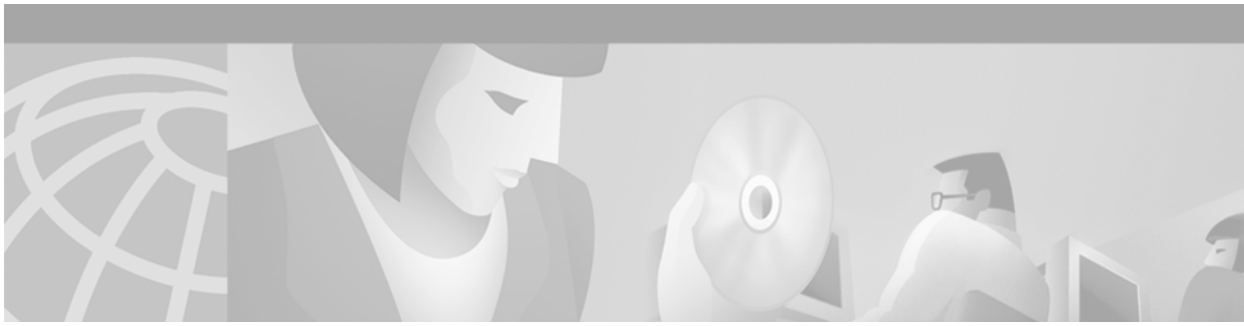
The following examples show TACACS+ user profiles that could be used for selective virtual access interface creation.

This example shows AAA per-user configuration for a TACACS+ user profile:

```
user = foo {
                name = "foo"
                global = cleartext test
                service = PPP protocol= ip {
                        inacl#1="deny 10.10.10.10 0.0.0.0"
                        inacl#1="permit any"
                }
        }
```

This example shows a virtual profile AAA configuration for a TACACS+ user profile:

```
TACACS+ user profile:
        user = foo {
                name = "foo"
                global = cleartext test
                service = PPP protocol= lcp {
                        interface-config="keepalive 30\nppp max-bad-auth 4"
                }
                service = ppp protocol = ip {
                }
        }
```



Configuring Virtual Profiles

This chapter describes how to configure virtual profiles for use with virtual access interfaces. It includes the following main sections:

Virtual profiles run on all Cisco IOS platforms that support Multilink PPP (MLP).

We recommend that unnumbered addresses be used in virtual template interfaces to ensure that duplicate network addresses are not created on virtual access interfaces.

Virtual profiles interoperate with Cisco dial-on-demand routing (DDR), MLP, and dialers such as ISDN.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the virtual profile commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Virtual Profiles Overview

A virtual profile is a unique application that can create and configure a virtual access interface dynamically when a dial-in call is received and that can tear down the interface dynamically when the call ends. Virtual profiles support these encapsulation methods:

- PPP
- MLP
- High-Level Data Link Control (HDLC)
- Link Access Procedure, Balanced (LAPB)
- X.25
- Frame Relay

Any commands for these encapsulations that can be configured under a serial interface can be configured under a virtual profile stored in a user file on an authentication, authorization, and accounting (AAA) server and a virtual profile virtual template configured locally. The AAA server daemon downloads them as text to the network access server and is able to handle multiple download attempts.

The configuration information for a virtual profiles virtual access interface can come from a virtual template interface or from user-specific configuration stored on a AAA server, or both.

If a B interface is bound by the calling line identification (CLID) to a created virtual access interface cloned from a virtual profile or a virtual template interface, only the configuration from the virtual profile or the virtual template takes effect. The configuration on the D interface is ignored unless successful binding occurs by PPP name. Both the link and network protocols run on the virtual access interface instead of the B channel, unless the encapsulation is PPP.

Moreover, in previous releases of Cisco IOS software, downloading a profile from an AAA server and creating and cloning a virtual access interface was always done after the PPP call answer and link control protocol (LCP) up processes. The AAA download is part of authorization. But in the current release, these operations must be performed before the call is answered and the link protocol goes up. This restriction is a new AAA nonauthenticated authorization step. The virtual profile code handles multiple download attempts and identifies whether a virtual access interface was cloned from a downloaded virtual profile.

When a successful download is done through nonauthenticated authorization and the configuration on the virtual profile has encapsulation PPP and PPP authentication, authentication is negotiated as a separate step after LCP comes up.

The per-user configuration feature also uses configuration information gained from a AAA server. However, per-user configuration uses *network* configurations (such as access lists and route filters) downloaded during Network Control Protocol (NCP) negotiations.

Two rules govern virtual access interface configuration by virtual profiles, virtual template interfaces, and AAA configurations:

- Each virtual access application can have at most one template to clone from but can have multiple AAA configurations to clone from (virtual profiles AAA information and AAA per-user configuration, which in turn might include configuration for multiple protocols).
- When virtual profiles are configured by virtual template, its template has higher priority than any other virtual template.

See the section "How Virtual Profiles Work—Four Configuration Cases" for a description of the possible configuration sequences for configuration by virtual template or AAA or both. See the section "Multilink PPP Effect on Virtual Access Interface Configuration" for a description of the possible configuration sequences that depend on the presence or absence by MLP or another virtual access feature that clones a virtual template interface.

DDR Configuration of Physical Interfaces

Virtual profiles fully interoperate with physical interfaces in the following DDR configuration states when no other virtual access interface application is configured:

- Dialer profiles are configured for the interface—The dialer profile is used instead of the virtual profiles configuration.
- DDR is not configured on the interface—Virtual profiles overrides the current configuration.
- Legacy DDR is configured on the interface—Virtual profiles overrides the current configuration.

Note If a dialer interface is used (including any ISDN dialer), its configuration is used on the physical interface instead of the virtual profiles configuration.

Multilink PPP Effect on Virtual Access Interface Configuration

As shown in Table 28, exactly how a virtual access interface will be configured depends on the following three factors:

- Whether virtual profiles are configured by a virtual template, by AAA, by both, or by neither. In the table, these states are shown as "VP VT only," "VP AAA only," "VP VT and VP AAA," and "No VP at all," respectively.
- The presence or absence of a dialer interface.
- The presence or absence of MLP. The column label "MLP" is a stand-in for any virtual access feature that supports MLP and clones from a virtual template interface.

In Table 28, "(Multilink VT)" means that a virtual template interface is cloned *if* one is defined for MLP or a virtual access feature that uses MLP.

Table 28 Virtual Profiles Configuration Cloning Sequence

Virtual Profiles Configuration	MLP No Dialer	MLP Dialer	No MLP No Dialer	No MLP Dialer
VP VT only	VP VT	VP VT	VP VT	VP VT
VP AAA only	(Multilink VT) VP AAA	(Multilink VT) VP AAA	VP AAA	VP AAA
VP VT and VP AAA	VP VT VP AAA	VP VT VP AAA	VP VT VP AAA	VP VT VP AAA
No VP at all	(Multilink VT)[1]	Dialer[2]	No virtual access interface is created.	No virtual access interface is created.

1. The multilink bundle virtual access interface is created and uses the default settings for MLP or the relevant virtual access feature that uses MLP.
2. The multilink bundle virtual access interface is created and cloned from the dialer interface configuration.

The order of items in any cell of the table is important. Where VP VT is shown above VP AAA, it means that first the virtual profile virtual template is cloned on the interface, and then the AAA interface configuration for the user is applied to it. The user-specific AAA interface configuration adds to the configuration and overrides any conflicting physical interface or virtual template configuration commands.

Interoperability with Other Features That Use Virtual Templates

Virtual profiles also interoperate with virtual access applications that clone a virtual template interface. Each virtual access application can have at most one template to clone from but can clone from multiple AAA configurations.

The interaction between virtual profiles and other virtual template applications is as follows:

- If virtual profiles are enabled and a virtual template is defined for it, the virtual profile virtual template is used.
- If virtual profiles are configured by AAA alone (no virtual template is defined for virtual profiles), the virtual template for another virtual access application (virtual private dialup networks or VPDNs, for example) can be cloned onto the virtual access interface.
- A virtual template, if any, is cloned to a virtual access interface before the virtual profiles AAA configuration or AAA per-user configuration. AAA per-user configuration, if used, is applied last.

How Virtual Profiles Work—Four Configuration Cases

This section describes virtual profiles and the various ways that they can work with virtual template interfaces, user-specific AAA interface configuration, and MLP or another feature that requires MLP.

Virtual profiles separate configuration information into two logical parts:

- Generic—Common configuration for dial-in users plus other router-dependent configuration. This common and router-dependent information can define a virtual template interface stored locally on the router. The generic virtual template interface is independent of and can override the configuration of the physical interface on which a user dialed in.
- User-specific interface information—Interface configuration stored in a user file on an AAA server; for example, the authentication requirements and specific interface settings for a specific user. The settings are sent to the router in the response to the request from the router to authenticate the user, and the settings can override the generic configuration. This process is explained more in the section "Virtual Profiles Configured by AAA" later in this chapter.

These logical parts can be used separately or together. Four separate cases are possible:

- Case 1: Virtual Profiles Configured by Virtual Template—Applies the virtual template.
- Case 2: Virtual Profiles Configured by AAA—Applies the user-specific interface configuration received from the AAA server.
- Case 3: Virtual Profiles Configured by Virtual Template and AAA Configuration—Applies the virtual template and the user-specific interface configuration received from the AAA server.
- Case 4: Virtual Profiles Configured by AAA, and a Virtual Template Defined by Another Application—Applies the other application's virtual template interface and then applies the user-specific interface configuration received from the AAA server.

Note All cases assume that AAA is configured globally on the router, that the user has configuration information in the user file on the AAA server, that PPP authentication and authorization proceed as usual, and that the AAA server sends user-specific configuration information in the authorization approval response packet to the router.

The cases also assume that AAA works as designed and that the AAA server sends configuration information for the dial-in user to the router, even when virtual profiles by virtual template are configured.

See the sections "Virtual Profiles Configured by Virtual Templates," "Virtual Profiles Configured by AAA Configuration," "Virtual Profiles Configured by Virtual Templates and AAA Configuration," and "Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway" later in this chapter for examples of how to configure these cases.

Case 1: Virtual Profiles Configured by Virtual Template

In the case of virtual profiles configured by virtual template, the software functions as follows:

- If the physical interface is configured for dialer profiles (a DDR feature), the router looks for a dialer profile for the specific user.
- If a dialer profile is found, it is used instead of virtual profiles.
- If a dialer profile is not found for the user, or legacy DDR is configured, or DDR is not configured at all, virtual profiles create a virtual access interface for the user.

The router applies the configuration commands that are in the virtual template interface to create and configure the virtual profile. The template includes generic interface information and router-specific information, but no user-specific information. No matter whether a user dialed in on a synchronous serial, an asynchronous serial, or an ISDN interface, the dynamically created virtual profile for the user is configured as specified in the virtual template.

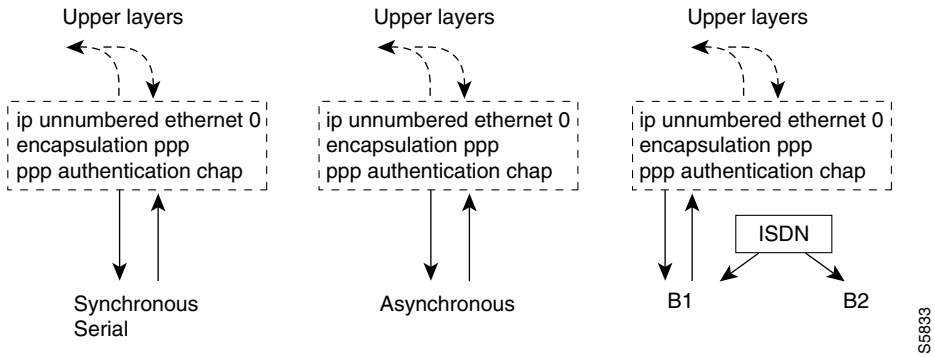
Then the router interprets the lines in the AAA authorization approval response from the server as Cisco IOS commands to apply to the virtual profile for the user.

Data flows through the virtual profile, and the higher layers treat it as the interface for the user.

For example, if a virtual template included only the three commands **ip unnumbered ethernet 0**, **encapsulation ppp,** and **ppp authentication chap**, the virtual profile for any dial-in user would include those three commands.

In Figure 68, the dotted box represents the virtual profile configured with the commands that are in the virtual template, no matter which interface the call arrives on.

Figure 68 Virtual Profiles by Virtual Template



See the section "Configuring Virtual Profiles by Virtual Template" later in this chapter for configuration tasks for this case.

Case 2: Virtual Profiles Configured by AAA

In this case, no dialer profile (a DDR feature) is defined for the specific user and no virtual template for virtual profiles is defined, but virtual profiles by AAA are enabled on the router.

During the PPP authorization phase for the user, the AAA server responds as usual to the router. The authorization approval contains configuration information for the user. The router interprets each of the lines in the AAA response from the server as Cisco IOS commands to apply to the virtual profile for the user.

Note If MLP is negotiated, the MLP virtual template is cloned first (this is the second row), and then interface-specific commands included in the AAA response from the server for the user are applied. The MLP virtual template overrides any conflicting interface configuration, and the AAA interface configuration overrides any conflicting configuration from both the physical interface and the MLP virtual template.

The router applies all the user-specific interface commands received from the AAA server.

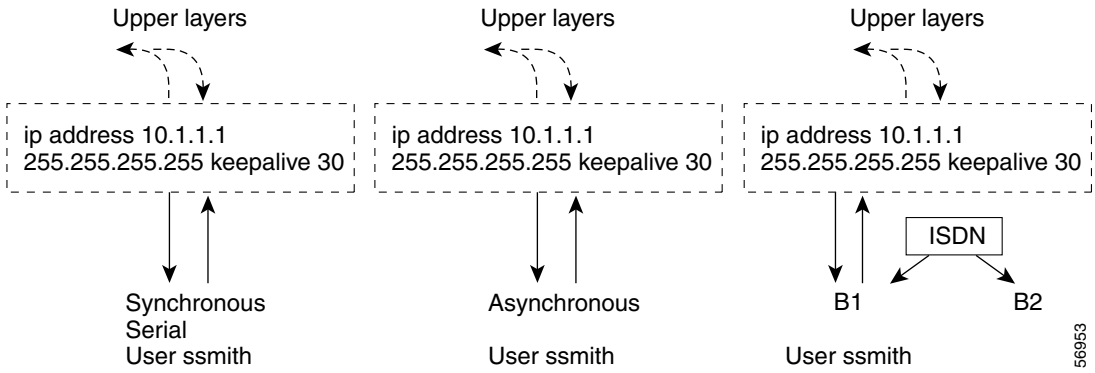
Suppose, for example, that the router interpreted the response by the AAA server as including only the following two commands for this user:

```
ip address 10.10.10.10 255.255.255.255
keepalive 30
```

In Figure 69, the dotted box represents the virtual profile configured only with the commands received from the AAA server, no matter which interface the incoming call arrived on. On the AAA RADIUS server, the attribute-value (AV) pair might have read as follows, where "\n" means to start a new command line:

```
cisco-avpair = "lcp:interface-config=ip address 10.10.10.10 255.255.255.0\nkeepalive 30",
```

Figure 69 Virtual Profiles by AAA Configuration



See the section "Configuring Virtual Profiles by AAA Configuration" later in this chapter for configuration tasks for this case.

Case 3: Virtual Profiles Configured by Virtual Template and AAA Configuration

In this case, no DDR dialer profile is defined for the specific user, a virtual template for virtual profiles is defined, virtual profiles by AAA is enabled on the router, the router is configured for AAA, and a user-specific interface configuration for the user is stored on the AAA server.

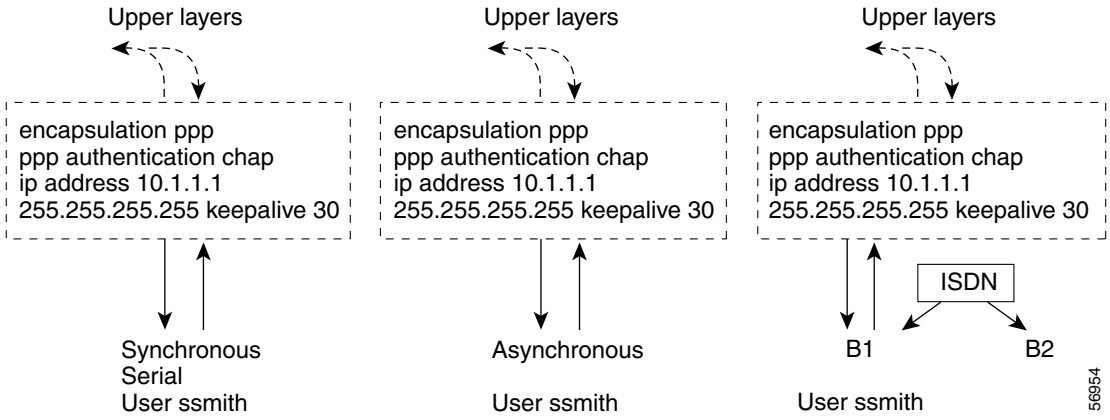
The router performs the following tasks in order:

1. Dynamically creates a virtual access interface cloned from the virtual template defined for virtual profiles.
2. Applies the user-specific interface configuration received from the AAA server.

If any command in the user's configuration conflicts with a command on the original interface or a command applied by cloning the virtual template, the user-specific command overrides the other command.

Suppose that the router had the virtual template as defined in Case 1 and the AAA user configuration as defined in Case 2. In Figure 70 the dotted box represents the virtual profile configured with configuration information from both sources, no matter which interface the incoming call arrived on. The **ip address** command has overridden the **ip unnumbered** command.

Figure 70 Virtual Profiles by Both Virtual Template and AAA Configuration



See the section "Configuring Virtual Profiles by Both Virtual Template and AAA Configuration" later in this chapter for configuration tasks for this case.

Case 4: Virtual Profiles Configured by AAA, and a Virtual Template Defined by Another Application

In this case, no DDR dialer profile is defined for the specific user, virtual profiles by AAA are configured on the router but no virtual template is defined for virtual profiles, and a user-specific interface configuration is stored on the AAA server. In addition, a virtual template is configured for some other virtual access application (a VPDN, for example).

The router performs the following tasks in order:

1. Dynamically creates a virtual access interface and clones the virtual template from the other virtual access application onto it.
2. Applies the user-specific interface configuration received from the AAA server.

If any command in the virtual template conflicts with a command on the original interface, the template overrides it.

If any command in the AAA interface configuration for the user conflicts with a command in the virtual template, the user AAA interface configuration conflicts will override the virtual template.

If per-user configuration is also configured on the AAA server, that network protocol configuration is applied to the virtual access interface last.

The result is a virtual interface unique to that user.

How to Configure Virtual Profiles

To configure virtual profiles for dial-in users, perform the tasks in *one* of the first three sections and then troubleshoot the configuration by performing the tasks in the last section:

- Configuring Virtual Profiles by Virtual Template (As required)
- Configuring Virtual Profiles by AAA Configuration (As required)
- Configuring Virtual Profiles by Both Virtual Template and AAA Configuration (As required)
- Troubleshooting Virtual Profile Configurations (As required)

Note Do not define a DDR dialer profile for a user if you intend to define virtual profiles for the user.

See the section "Configuration Examples for Virtual Profiles" at the end of this chapter for examples of how to use virtual profiles in your network configuration.

Configuring Virtual Profiles by Virtual Template

To configure virtual profiles by virtual template, complete these two tasks:

- Creating and Configuring a Virtual Template Interface
- Specifying a Virtual Template Interface for Virtual Profiles

Note The order in which these tasks is performed is not crucial. However, both tasks must be completed before virtual profiles are used.

Creating and Configuring a Virtual Template Interface

Because a virtual template interface is a serial interface, all the configuration commands that apply to serial interfaces can also be applied to virtual template interfaces, except **shutdown** and **dialer** commands.

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered ethernet 0`	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template interface.

Other optional PPP configuration commands can be added to the virtual template configuration. For example, you can add the **ppp authentication chap** command.

Specifying a Virtual Template Interface for Virtual Profiles

To specify a virtual template interface as the source of information for virtual profiles, use the following command in global configuration mode:

Command	Purpose
`Router(config)# virtual-profile virtual-template` *number*	Specifies the virtual template interface as the source of information for virtual profiles.

Virtual template numbers range from 1 to 25.

Configuring Virtual Profiles by AAA Configuration

To configure virtual profiles by AAA only, complete these three tasks in any order. All tasks must be completed before virtual profiles are used.

- On the AAA server, create user-specific interface configurations for each of the specific users to use this method. See your AAA server documentation for more detailed configuration information about your AAA server.
- Configure AAA on the router, as described in the *Cisco IOS Security Configuration Guide*, Release 12.2.
- Specify AAA as the source of information for virtual profiles.

To specify AAA as the source of information for virtual profiles, use the following command in global configuration mode:

Command	Purpose
`Router(config)# virtual-profile aaa`	Specifies AAA as the source of user-specific interface configuration.

If you also want to use per-user configuration for network protocol access lists or route filters for individual users, see the chapter "Configuring Per-User Configuration" in this publication. In this case, no virtual template interface is defined for virtual profiles.

Configuring Virtual Profiles by Both Virtual Template and AAA Configuration

Use of user-specific AAA interface configuration information with virtual profiles requires the router to be configured for AAA and requires the AAA server to have user-specific interface configuration AV-pairs. The relevant AV-pairs (on a RADIUS server) begin as follows:

```
cisco-avpair = "lcp:interface-config=...",
```

The information that follows the equal sign (=) could be any Cisco IOS interface configuration command. For example, the line might be the following:

```
cisco-avpair = "lcp:interface-config=ip address 192.168.200.200 255.255.255.0",
```

Use of a virtual template interface with virtual profiles requires a virtual template to be defined specifically for virtual profiles.

To configure virtual profiles by both virtual template interface and AAA configuration, complete the following tasks in any order. All tasks must be completed before virtual profiles are used.

- On the AAA server, create user-specific interface configurations for each of the specific users to use this method. See your AAA server documentation for more detailed configuration information about your AAA server.
- Configure AAA on the router, as described in the *Cisco IOS Security Configuration Guide* publication.
- Creating and Configuring a Virtual Template Interface, described later in this chapter.
- Specifying Virtual Profiles by Both Virtual Templates and AAA, described later in this chapter.

Creating and Configuring a Virtual Template Interface

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered ethernet 0`	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template interface.

Because the software treats a virtual template interface as a serial interface, all the configuration commands that apply to serial interfaces can also be applied to virtual template interfaces, except **shutdown** and **dialer** commands. Other optional PPP configuration commands can also be added to the virtual template configuration. For example, you can add the **ppp authentication chap** command.

Specifying Virtual Profiles by Both Virtual Templates and AAA

To specify both the virtual template interface and the AAA per-user configuration as sources of information for virtual profiles, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# virtual-profile virtual-template` *number*	Defines the virtual template interface as the source of information for virtual profiles.
Step 2	`Router(config)# virtual-profile aaa`	Specifies AAA as the source of user-specific configuration for virtual profiles.

If you also want to use per-user configuration for network protocol access lists or route filters for individual users, see the chapter “Configuring Per-User Configuration” in this publication.

Troubleshooting Virtual Profile Configurations

To troubleshoot the virtual profiles configurations, use any of the following **debug** commands in EXEC mode:

Command	Purpose
`Router# debug dialer`	Displays information about dial calls and negotiations and virtual profile events.
`Router# debug aaa per-user`	Displays information about the per-user configuration downloaded from the AAA server.
`Router# debug vtemplate`	Displays cloning information for a virtual access interface from the time it is cloned from a virtual template to the time it comes down.

Configuration Examples for Virtual Profiles

The following sections provide examples for the four cases described in this chapter:

- Virtual Profiles Configured by Virtual Templates
- Virtual Profiles Configured by AAA Configuration
- Virtual Profiles Configured by Virtual Templates and AAA Configuration
- Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway

In these examples, BRI 0 is configured for legacy DDR, and interface BRI 1 is configured for dialer profiles. Note that interface dialer 0 is configured for legacy DDR. Interface dialer 1 is a dialer profile.

The intention of the examples is to show how to configure virtual profiles. In addition, the examples show the interoperability of DDR and dialer profiles in the respective cases with various forms of virtual profiles.

The same user names (John and Rick) occur in all these examples. Note the different configuration allowed to them in each of the four examples.

John is a normal user and can dial in to BRI 0 only. Rick is a privileged user who can dial in to BRI 0 and BRI 1. If Rick dials into BRI 1, the dialer profile will be used. If Rick dials into BRI 0, virtual profiles will be used. Because John does not have a dialer profile, only virtual profiles can be applied to John.

To see an example of a configuration using virtual profiles and the Dynamic Multiple Encapsulations feature, see the "Multiple Encapsulations over ISDN" example in the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles."

Virtual Profiles Configured by Virtual Templates

The following example shows a router configured for virtual profiles by virtual template. (Virtual profiles do not have any interface-specific AAA configuration.) Comments in the example draw attention to specific features or ignored lines.

In this example, the same virtual template interface applies to both users; they have the same interface configurations.

Router Configuration

```
! Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
! The following command is required.
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
! Specify configuration of virtual profiles by virtual template.
! This is the key command for this example.
virtual-profile virtual-template 1
!
! Define the virtual template.
interface Virtual-Template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp authentication chap
!
switch-type basic-dms100
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
interface BRI 1
 description Connected to 104
 encapsulation ppp
! Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR for John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
! Enable legacy DDR.
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 1 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
! Disable fast switching.
 no ip route-cache
 ppp authentication chap
 dialer-list 1 protocol ip permit
```

Virtual Profiles Configured by AAA Configuration

The following example shows the router configuration for virtual profiles by AAA and the AAA server configuration for user-specific interface configurations. John and Rick have different IP addresses.

In the AAA configuration cisco-avpair lines, "\n" is used to indicate the start of a new Cisco IOS command line.

AAA Configuration for John and Rick

```
john Password = "welcome"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 75\nip address 172.16.100.100
 255.255.255.0",
rick Password = "emoclew"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 100\nip address 192.168.200.200
 255.255.255.0"
```

Router Configuration

```
! Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
! This is a key command for this example.
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
! Specify configuration of virtual profiles by aaa.
! This is a key command for this example.
virtual-profiles aaa
!
! Interface BRI 0 is configured for legacy DDR.
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
! Interface BRI 1 is configured for dialer profiles.
interface BRI 1
 description Connected to 104
 encapsulation ppp
! Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR for John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
! Enable legacy DDR.
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
```

```
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 1 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
! Disable fast switching.
 no ip route-cache
 ppp authentication chap
 dialer-list 1 protocol ip permit
```

Virtual Profiles Configured by Virtual Templates and AAA Configuration

The following example shows how virtual profiles can be configured by both virtual templates and AAA configuration. John and Rick can dial in from anywhere and have their same keepalive settings and their own IP addresses.

The remaining AV pair settings are not used by virtual profiles. They are the network protocol access lists and route filters used by AAA-based per-user configuration.

In the AAA configuration cisco-avpair lines, "\n" is used to indicate the start of a new Cisco IOS command line.

AAA Configuration for John and Rick

```
john Password = "welcome"
     User-Service-Type = Framed-User,
     Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=keepalive 75\nip address 10.16.100.100
255.255.255.0",
        cisco-avpair = "ip:rte-fltr-out#0=router igrp 60",
        cisco-avpair = "ip:rte-fltr-out#3=deny 172.16.0.0 0.255.255.255",
        cisco-avpair = "ip:rte-fltr-out#4=deny 172.17.0.0 0.255.255.255",
        cisco-avpair = "ip:rte-fltr-out#5=permit any"
rick Password = "emoclew"
     User-Service-Type = Framed-User,
     Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=keepalive 100\nip address 192.168.200.200
255.255.255.0",
        cisco-avpair = "ip:inacl#3=permit ip any any precedence immediate",
        cisco-avpair = "ip:inacl#4=deny igrp 10.0.1.2 255.255.0.0 any",
        cisco-avpair = "ip:outacl#2=permit ip any any precedence immediate",
        cisco-avpair = "ip:outacl#3=deny igrp 10.0.9.10 255.255.0.0 any"
```

Router Configuration

```
! Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
! This is a key command for this example.
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
```

```
! Specify use of virtual profiles and a virtual template.
! The following two commands are key for this example.
virtual-profile virtual-template 1
virtual-profile aaa
!
! Define the virtual template.
interface Virtual-Template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp authentication chap
!
! Interface BRI 0 is configured for legacy DDR.
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
! Interface BRI 1 is configured for dialer profiles.
interface BRI 1
 description Connected to 104
 encapsulation ppp
! Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR to dial out to John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
! Disable fast switching.
 no ip route-cache
 ppp authentication chap
!
 dialer-list 1 protocol ip permit
```

Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway

Like the virtual profiles configured by AAA example earlier in this section, the following example shows the router configuration for virtual profiles by AAA. The user file on the AAA server also includes interface configuration for John and Rick, the two users. Specifically, John and Rick each have their own IP addresses when they are in privileged mode.

In this case, however, the router is also configured as the VPDN home gateway. It clones the VPDN virtual template interface first and then clones the virtual profiles AAA interface configuration. If per-user configuration were configured on this router and the user file on the AAA server had network protocol information for the two users, that information would be applied to the virtual access interface last.

In the AAA configuration cisco-avpair lines, "\n" is used to indicate the start of a new Cisco IOS command line.

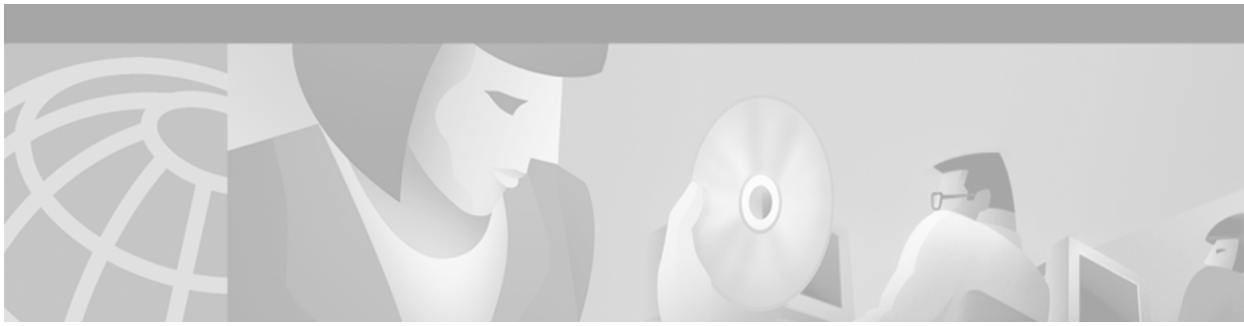
AAA Configuration for John and Rick

```
john Password = "welcome"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 75\nip address 10.100.100.100
 255.255.255.0",
rick Password = "emoclew"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 100\nip address 192.168.200.200
 255.255.255.0"
```

Router Configuration

```
!Configure the router as the VPDN home gateway.
!
!Enable VPDN and specify the VPDN virtual template to use on incoming calls from the
!network access server.
vpdn enable
vpdn incoming dallas_wan go_blue virtual-template 6
!
!Configure the virtual template interface for VPDN.
interface virtual template 6
ip unnumbered ethernet 0
encapsulation ppp
ppp authentication chap
!
!Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
!Specify configuration of virtual profiles by aaa.
virtual-profiles aaa
!
!Configure the physical synchronous serial 0 interface.
interface Serial 0
 description Connected to 101
 encapsulation ppp
```

```
!Disable fast switching.
 no ip route-cache
 ppp authentication chap
!
!Configure serial interface 1 for DDR. S1 uses dialer rotary group 0, which is
!defined on BRI interface 0.
interface serial 1
 description Connected to 102
 encapsulation ppp
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
interface BRI 1
 description Connected to 104
 encapsulation ppp
!Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
!Configure dialer interface 0 for DDR to call and receive calls from John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
!Enable legacy DDR.
 dialer in-band
!Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
!Configure dialer interface 1 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
!Disable fast switching.
 no ip route-cache
 ppp authentication chap
 dialer-list 1 protocol ip permit
```



Configuring Virtual Private Networks

This chapter describes how to configure, verify, maintain, and troubleshoot a Virtual Private Network (VPN). It includes the following main sections:

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

VPN Technology Overview

A VPN carries private data over a public network and extends remote access to users over a shared infrastructure. VPNs maintain the same security and management policies as a private network. They are the most cost-effective method of establishing a point-to-point connection between remote users and a central network.

A benefit of VPNs or, more appropriately, access VPNs, is the way they delegate responsibilities for the network. The customer outsources the responsibility for the information technology (IT) infrastructure to an Internet service provider (ISP) that maintains the modems that the remote users dial in to (called modem pools), the access servers, and the internetworking expertise. The customer is then only responsible for authenticating its users and maintaining its network.

Instead of connecting directly to the network by using the expensive Public Switched Telephone Network (PSTN), access VPN users need only use the PSTN to connect to the ISP local point of presence (POP). The ISP then uses the Internet to forward users from the POP to the customer network. Forwarding a user call over the Internet provides dramatic cost savings for the customer. Access VPNs use Layer 2 tunneling technologies to create a virtual point-to-point connection between users and the

customer network. These tunneling technologies provide the same direct connectivity as the expensive PSTN by using the Internet. This means that users anywhere in the world have the same connectivity as they would at the customer headquarters.

VPNs allow separate and autonomous protocol domains to share common access infrastructure including modems, access servers, and ISDN routers. VPNs use the following tunneling protocols to tunnel link-level frames:

- Layer 2 Forwarding (L2F)
- Layer 2 Tunneling Protocol (L2TP)
- Point-to-Point Tunneling Protocol (PPTP)

Using one of these protocols, an ISP or other access service can create a virtual tunnel to link customer remote sites or remote users with corporate home networks. In particular, a network access server (NAS) at the ISP POP exchanges PPP messages with the remote users and communicates by L2F, L2TP, or PPTP requests and responses with the customer tunnel server to set up tunnels.

L2F, L2TP, and PPTP pass protocol-level packets through the virtual tunnel between endpoints of a point-to-point connection.

Frames from the remote users are accepted by the ISP POP, stripped of any linked framing or transparency bytes, encapsulated in L2F, L2TP or PPTP, and forwarded over the appropriate tunnel. The customer tunnel server accepts these frames, strips the Layer 2 encapsulation, and processes the incoming frames for the appropriate interface.

Cisco routers fast switch VPN traffic. In stack group environments in which some VPN traffic is offloaded to a powerful router, fast switching provides improved scalability.

VPDN MIB

The VPDN MIB offers a mechanism to track failures of user calls in a VPN system, allowing Simple Network Management Protocol (SNMP) retrieval of user call failure information, on a per-user basis.

Refer to the Cisco VPDN Management MIB for a list of supported objects for the VPDN MIB.

VPN Hardware Terminology

As new tunneling protocols have been developed for VPNs, new terminology has been created to describe the hardware involved in VPNs. Fundamentally, two routers are needed for a VPN:

- Network access server (NAS)—It receives incoming calls for dial-in VPNs and places outgoing calls for dial-out VPNs. Typically it is maintained by an ISP that wishes to provide VPN services to its customers.
- Tunnel server—It terminates dial-in VPNs and initiates dial-out VPNs. Typically it is maintained by the ISP customer and is the contact point for the customer network.

For the sake of clarity, we will use these generic terms, and not the technology-specific terms. Table 29 lists the generic terms ant the technology-specific terms that are often used for these devices.

Table 29 VPN Hardware Terminology

Generic Term	L2F Term	L2TP Term	PPTP Term
Tunnel Server	Home Gateway	L2TP Network Server (LNS)	PPTP Network Server (PNS)
Network Access Server (NAS)	NAS	L2TP Access Concentrator (LAC)	PPTP Access Concentrator (PAC)

In dial-in scenarios, users dial in to the NAS, and the NAS forwards the call to the tunnel server using a VPN tunnel.

In dial-out scenarios, the tunnel server initiates a VPN tunnel to the NAS, and the NAS dials out to the clients.

VPN Architectures

VPNs are designed on the basis of one of two architectural options:

- Client-Initiated VPNs
- NAS-Initiated VPNs

Client-Initiated VPNs

Users establish a tunnel across the ISP shared network to the customer network without an intermediate NAS participating in the tunnel negotiation and establishment. The customer manages the client software that initiates the tunnel. The main advantage of client-initiated VPNs is that they secure the connection between the client and the ISP. However, client-initiated VPNs are not as scalable and are more complex than NAS-initiated VPNs.

Client-initiated VPNs are also referred to as voluntary tunneling.

NAS-Initiated VPNs

Users dial in to the ISP NAS, which establishes a tunnel to the private network. NAS-initiated VPNs are more robust than client-initiated VPNs and do not require the client to maintain the tunnel-creating software. NAS-initiated VPNs do not encrypt the connection between the client and the ISP, but this is not a concern for most customers because the PSTN is much more secure than the Internet.

NAS-initiated VPNs are also referred to as compulsory tunneling.

Note In Cisco's VPN implementation, PPTP tunnels are client-initiated while L2F and L2TP tunnels are NAS-initiated.

PPTP Dial-In with MPPE Encryption

PPTP is a network protocol that enables the secure transfer of data from a remote client to a private enterprise server by creating a VPN across TCP/IP-based data networks. PPTP supports on-demand, multiprotocol, virtual private networking over public networks, such as the Internet.

Cisco supports client-initiated VPNs using PPTP. Therefore only the client and the tunnel server need to be configured. The client first establishes basic connectivity by dialing in to an ISP. Once the client has established a PPP session, it initiates a PPTP tunnel to the tunnel server. The tunnel server is configured to terminate PPTP tunnels and clone virtual-access interfaces from virtual templates.

Microsoft Point-to-Point Encryption (MPPE) is an outcropping technology that can be used to encrypt PPTP VPNs. It encrypts the entire session from the client to the tunnel server.

This section describes the following aspects of PPTP and MPPE:

PPTP Tunnel Negotiation

The following describes the protocol negotiation events that establish a PPTP tunnel:

1. The client dials in to the ISP and establishes a PPP session.
2. The client establishes a TCP connection with the tunnel server.
3. The tunnel server accepts the TCP connection.
4. The client sends a PPTP SCCRQ message to the tunnel server.
5. The tunnel server establishes a new PPTP tunnel and replies with an SCCRP message.
6. The client initiates the session by sending an OCRQ message to the tunnel server.
7. The tunnel server creates a virtual-access interface.
8. The tunnel server replies with an OCRP message.

Flow Control Alarm

The flow control alarm is a new function that indicates if PPTP detects congestion or lost packets. When a flow control alarm goes off, PPTP reduces volatility and additional control traffic by establishing an accompanying stateful MPPE session.

For more information, see the **pptp flow-control static-rtt** command and the output from the **show vpdn session** command in the "Verifying a Client-Initiated VPN" section.

MPPE Overview

MPPE is an encryption technology developed by Microsoft to encrypt point-to-point links. These PPP connections can be over a dialup line or over a VPN tunnel. MPPE works as a subfeature of Microsoft Point-to-Point Compression (MPPC).

MPPC is a scheme used to compress PPP packets between Cisco and Microsoft client devices. The MPPC algorithm is designed to optimize bandwidth utilization in order to support multiple simultaneous connections.

MPPE is negotiated using bits in the MPPC option within the Compression Control Protocol (CCP) MPPC configuration option (CCP configuration option number 18).

MPPE uses the RC4 algorithm with either 40- or 128-bit keys. All keys are derived from the cleartext authentication password of the user. RC4 is stream cipher; therefore, the sizes of the encrypted and decrypted frames are the same size as the original frame. The Cisco implementation of MPPE is fully interoperable with that of Microsoft and uses all available options, including historyless mode. Historyless mode can increase throughput in lossy environments such as VPNs, because neither side needs to send CCP Resets Requests to synchronize encryption contexts when packets are lost.

MPPE Encryption Types

Two modes of MPPE encryption are offered:

- Stateful MPPE Encryption
- Stateless MPPE Encryption

Stateful MPPE Encryption

Stateful encryption provides the best performance but may be adversely affected by networks that experience substantial packet loss. If you choose stateful encryption, you should also configure flow control to minimize the detrimental effects of this lossiness.

Because of the way that the RC4 tables are reinitialized during stateful synchronization, it is possible that two packets may be encrypted using the same key. For this reason, stateful encryption may not be appropriate for lossy network environments (such as Layer 2 tunnels on the Internet).

Stateless MPPE Encryption

Stateless encryption provides a lower level of performance, but will be more reliable in a lossy network environment.

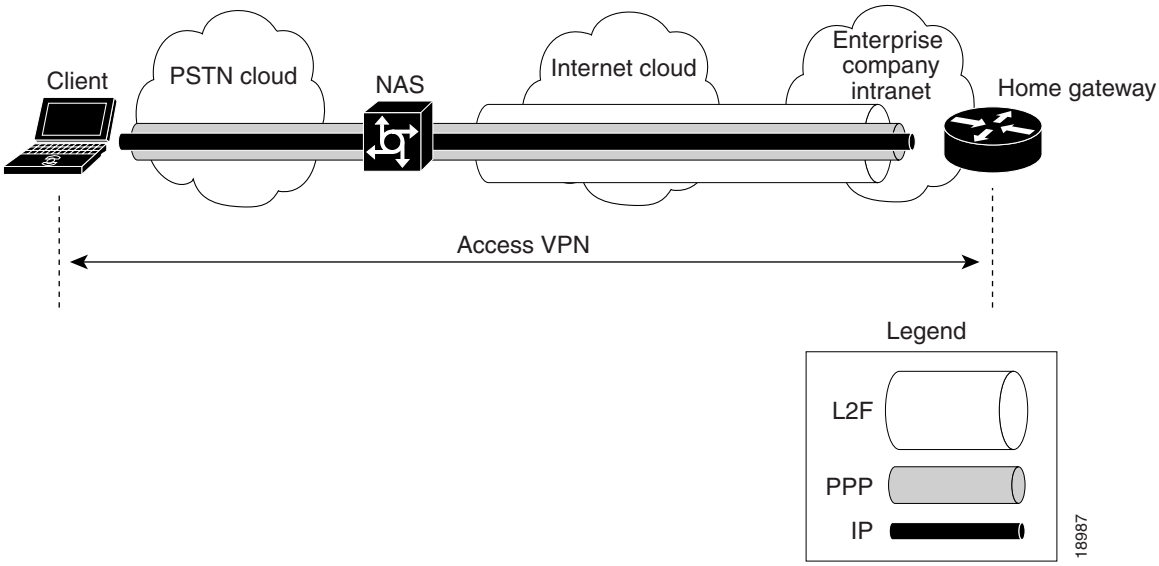
Caution If you choose stateless encryption, you *should not* configure flow control.

L2F Dial-In

VPNs use L2F or L2TP tunnels to tunnel the link layer of high-level protocols (for example, PPP frames or asynchronous High-Level Data Link Control (HDLC)). ISPs configure their NASs to receive calls from users and to forward the calls to the customer tunnel server. Usually, the ISP maintains only information about the tunnel server—the tunnel endpoint. The customer maintains the tunnel server users' IP addresses, routing, and other user database functions. Administration between the ISP and the tunnel server is reduced to IP connectivity.

Figure 71 shows the PPP link that runs between a client (the user hardware and software) and the tunnel server. The NAS and tunnel server establish an L2F tunnel that the NAS uses to forward the PPP link to the tunnel server. The VPN then extends from the client to the tunnel server. The L2F tunnel creates a virtual point-to-point connection between the client and the tunnel server.

Figure 71 End-to-End Access VPN Protocol Flow: L2F, PPP, and IP



The following sections give a functional description of the sequence of events that establish a VPN using L2F as the tunneling protocol:

- Protocol Negotiation Sequence
- L2F Tunnel Authentication Process

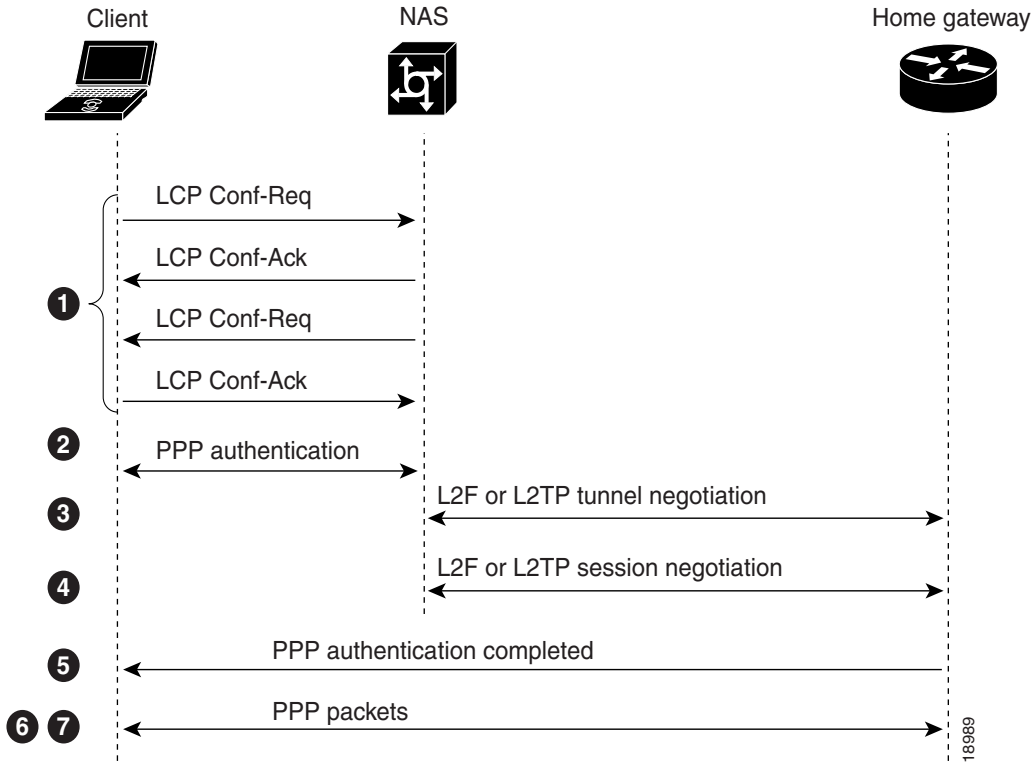
The "Protocol Negotiation Sequence" section provides an overview of the negotiation events that take place as the VPN is established. The "L2F Tunnel Authentication Process" section provides a detailed description of how the NAS and tunnel server establish the L2F tunnel.

Protocol Negotiation Sequence

A user who wants to connect to the customer tunnel server first establishes a PPP connection to the ISP NAS. The NAS then establishes an L2F tunnel with the tunnel server. Finally, the tunnel server authenticates the client username and password and establishes the PPP connection with the client.

Figure 72 shows the sequence of protocol negotiation events between the ISP NAS and the customer tunnel server.

Figure 72 Protocol Negotiation Events Between Access VPN Devices



The following explains the sequence of events shown in Figure 72:

1. The user client and the NAS conduct a standard PPP Link Control Protocol (LCP) negotiation.
2. The NAS begins PPP authentication by sending a Challenge Handshake Authentication Protocol (CHAP) challenge to the client.
3. The client replies with a CHAP response.
4. When the NAS receives the CHAP response, either the phone number that the user dialed in from (when using Dialed Number Information Service-based authentication) or the user domain name (when using authentication based on domain name) matches a configuration on either the NAS or its AAA server.

   This configuration instructs the NAS to create a VPN to forward the PPP session to the tunnel server by using an L2F tunnel.

   Because this is the first L2F session with the tunnel server, the NAS and the tunnel server exchange L2F_CONF packets, which prepare them to create the tunnel. Then they exchange L2F_OPEN packets, which open the L2F tunnel.
5. Once the L2F tunnel is open, the NAS and tunnel server exchange L2F session packets. The NAS sends an L2F_OPEN (Mid) packet to the tunnel server that includes the client information from the LCP negotiation, the CHAP challenge, and the CHAP response.

   The tunnel server forces this information on to a virtual access interface that it has created for the client and responds to the NAS with an L2F_OPEN (Mid) packet.
6. The tunnel server authenticates the CHAP challenge and response (using either local or remote AAA) and sends a CHAP Auth-OK packet to the client. This completes the three-way CHAP authentication.

7. When the client receives the CHAP Auth-OK packet, it can send PPP encapsulated packets to the tunnel server.

   The client and the tunnel server can now exchange I/O PPP encapsulated packets. The NAS acts as a transparent PPP frame forwarder.

   Subsequent PPP incoming sessions (designated for the same tunnel server) do not repeat the L2F tunnel negotiation because the L2F tunnel is already open.

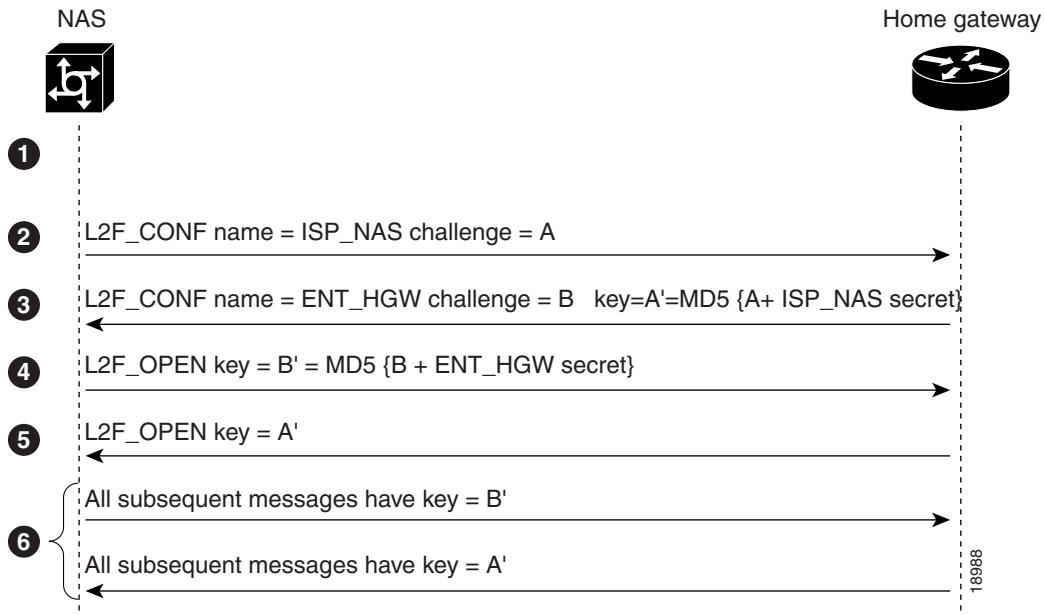
L2F Tunnel Authentication Process

When the NAS receives a call from a client that is to be tunneled to a tunnel server, it first sends a challenge to the tunnel server. The tunnel server then sends a combined challenge and response to the NAS. Finally, the NAS responds to the tunnel server challenge, and the two devices open the L2F tunnel.

Before the NAS and tunnel server can authenticate the tunnel, they must have a common "tunnel secret." A tunnel secret is a common shared secret that is configured on both the NAS and the tunnel server. For more information on tunnel secrets, see the "Configuring VPN Tunnel Authentication Using the L2TP Tunnel Password" section later in this chapter. By combining the tunnel secret with random value algorithms, which are used to encrypt the tunnel secret, the NAS and tunnel server authenticate each other and establish the L2F tunnel.

Figure 73 shows the tunnel authentication process.

Figure 73 L2F Tunnel Authentication Process



The following explains the sequence of events shown in Figure 73:

1. Before the NAS and tunnel server open an L2F tunnel, both devices must have a common tunnel secret in their configurations.
2. The NAS sends an L2F_CONF packet that contains the NAS name and a random challenge value, A.
3. After the tunnel server receives the L2F_CONF packet, it sends an L2F_CONF packet back to the NAS with the tunnel server name and a random challenge value, B. This message also includes a key containing A' (the MD5 of the NAS secret and the value A).
4. When the NAS receives the L2F_CONF packet, it compares the key A' with the MD5 of the NAS secret and the value A. If the key and value match, the NAS sends an L2F_OPEN packet to the tunnel server with a key containing B' (the Message Digest 5 (MD5) of the tunnel server secret and the value B).
5. When the tunnel server receives the L2F_OPEN packet, it compares the key B' with the MD5 of the tunnel server secret and the value B. If the key and value match, the tunnel server sends an L2F_OPEN packet to the NAS with the key A'.
6. All subsequent messages from the NAS include key = B'; all subsequent messages from the tunnel server include key = A'.

Once the tunnel server authenticates the client, the access VPN is established. The L2F tunnel creates a virtual point-to-point connection between the client and the tunnel server. The NAS acts as a transparent packet forwarder.

When subsequent clients dial in to the NAS, the NAS and tunnel server need not repeat the L2F tunnel negotiation because the L2F tunnel is already open.

L2TP Dial-In

L2TP is an emerging Internet Engineering Task Force (IETF) standard that combines the best features of two existing tunneling protocols: Cisco L2F (L2F) and Microsoft Point-to-Point Tunneling Protocol (PPTP).

L2TP offers the same full-range spectrum of features as L2F, but offers additional functionality. An L2TP-capable tunnel server will work with an existing L2F network access server and will concurrently support upgraded components running L2TP. Tunnel servers do not require reconfiguration each time an individual NAS is upgraded from L2F to L2TP. Table 30 offers a comparison of L2F and L2TP feature components.

Table 30 L2F and L2TP Feature Comparison

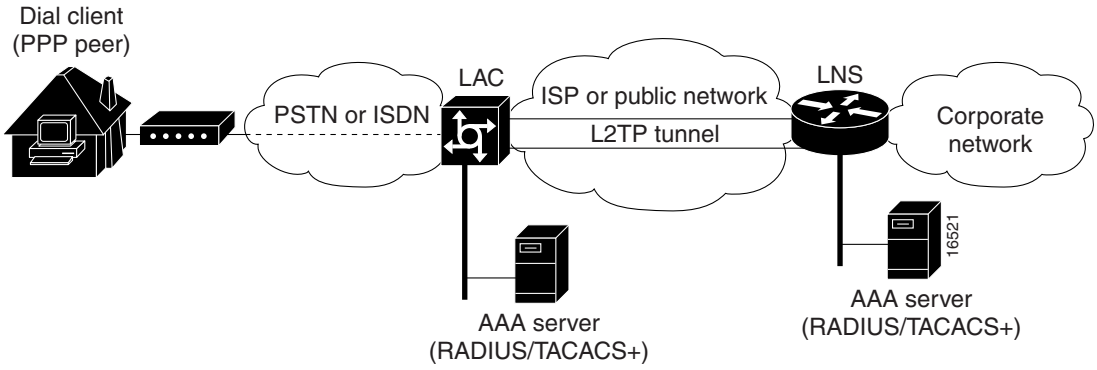
Function	L2F	L2TP
Flow Control	No	Yes
AVP hiding	No	Yes
Tunnel server load sharing	Yes	Yes
Tunnel server stacking/multihop support	Yes	Yes
Tunnel server primary and secondary backup	Yes	Yes
DNS name support	Yes	Yes
Domain name flexibility	Yes	Yes

Table 30 L2F and L2TP Feature Comparison (continued)

Function	L2F	L2TP
Idle and absolute timeout	Yes	Yes
Multilink PPP support	Yes	Yes
Multichassis Multilink PPP support	Yes	Yes
Security	• All security benefits of PPP, including multiple per-user authentication options (CHAP, MS-CHAP, PAP). • Tunnel authentication mandatory.	• All security benefits of PPP, including multiple per-user authentication options (CHAP, MS-CHAP, PAP). • Tunnel authentication optional.

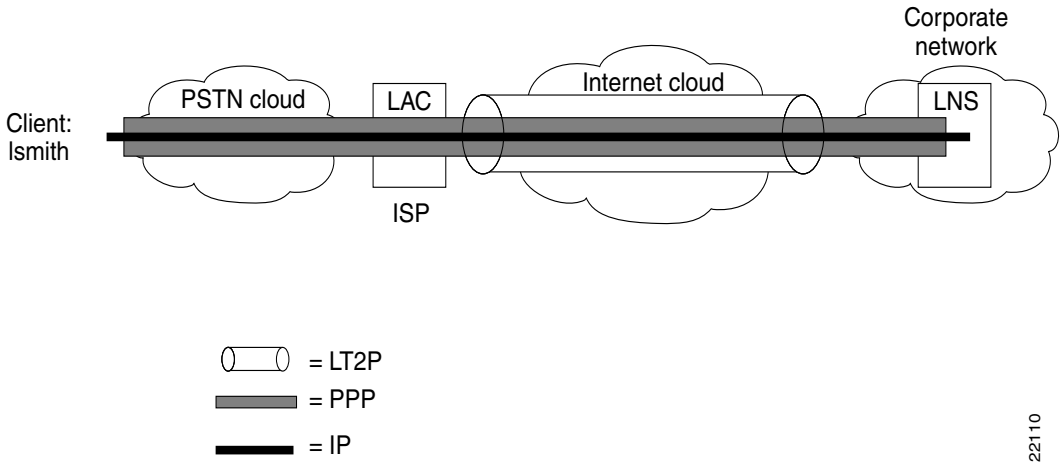
Traditional dialup networking services support only registered IP addresses, which limits the types of applications that are implemented over VPNs. L2TP supports multiple protocols and unregistered and privately administered IP addresses over the Internet. This allows the existing access infrastructure, such as the Internet, modems, access servers, and ISDN terminal adapters (TAs), to be used. It also allows customers to outsource dial-out support, thus reducing overhead for hardware maintenance costs and 800 number fees, and allows them to concentrate corporate gateway resources. Figure 74 shows the L2TP architecture in a typical dialup environment.

Figure 74 L2TP Architecture



The following sections supply additional detail about the interworkings and Cisco implementation of L2TP. Using L2TP tunneling, an Internet service provider (ISP) or other access service can create a virtual tunnel to link customer remote sites or remote users with corporate home networks. The NAS located at the POP of the ISP exchanges PPP messages with remote users and communicates by way of L2TP requests and responses with the customer tunnel server to set up tunnels. L2TP passes protocol-level packets through the virtual tunnel between endpoints of a point-to-point connection. Frames from remote users are accepted by the POP of the ISP, stripped of any linked framing or transparency bytes, encapsulated in L2TP and forwarded over the appropriate tunnel. The customer tunnel server accepts these L2TP frames, strips the L2TP encapsulation, and processes the incoming frames for the appropriate interface. Figure 75 shows the L2TP tunnel detail and how user “lsmith” connects to the tunnel server to access the designated corporate intranet.

Figure 75 L2TP Tunnel Structure



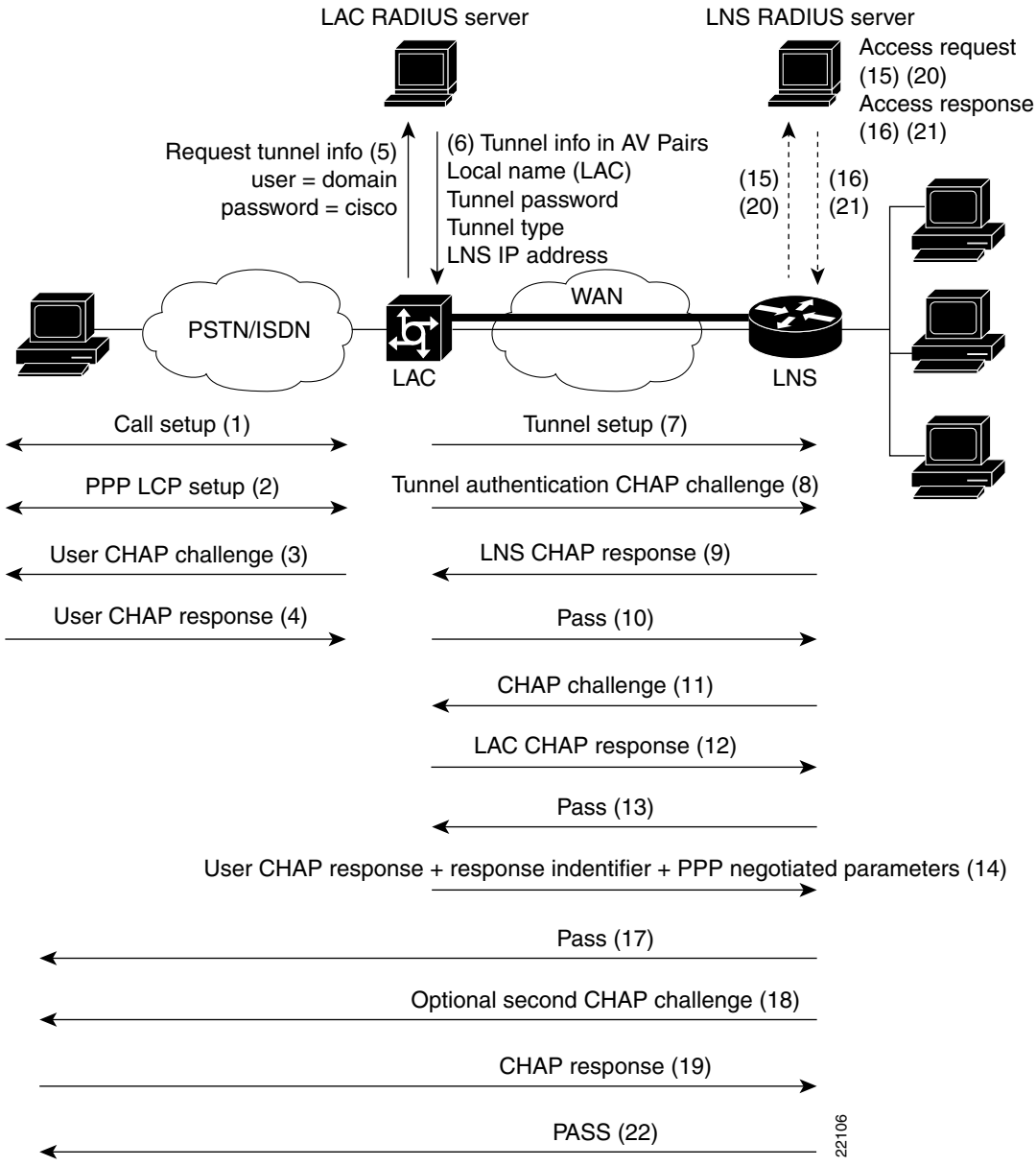
Incoming Call Sequence

The following describes the events required to establish a VPN connection between a remote user, a NAS at the ISP POP, and the tunnel server at the home LAN using an L2TP tunnel:

1. The remote user initiates a PPP connection to the ISP, using the analog telephone system or ISDN.
2. The ISP network NAS accepts the connection at the POP, and the PPP link is established.
3. After the end user and NAS negotiate LCP, the NAS partially authenticates the end user with CHAP or PAP. The username, domain name, or Dialed Number Information Service (DNIS) is used to determine whether the user is a VPN client. If the user is not a VPN client, authentication continues, and the client will access the Internet or other contacted service. If the username is a VPN client, the mapping will name a specific endpoint (the tunnel server).
4. The tunnel endpoints, the NAS, and the tunnel server authenticate each other before any sessions are attempted within a tunnel. Alternatively, the tunnel server can accept tunnel creation without any tunnel authentication of the NAS.
5. Once the tunnel exists, an L2TP session is created for the end user.
6. The NAS will propagate the LCP negotiated options and the partially authenticated CHAP/PAP information to the tunnel server. The tunnel server will funnel the negotiated options and authentication information directly to the virtual access interface. If the options configured on the virtual template interface do not match the negotiated options with the NAS, the connection will fail, and a disconnect will be sent to the NAS.

The result is that the exchange process appears to be between the dialup client and the remote tunnel server exclusively, as if no intermediary device (the NAS) is involved. Figure 76 offers a pictorial account of the L2TP incoming call sequence with its own corresponding sequence numbers. Note that the sequence numbers in Figure 76 are not related to the sequence numbers described in the previous table.

Figure 76 L2TP Incoming Call Flow



VPN Tunnel Authentication Search Order

When a call to a NAS is to be tunneled to a tunnel server, the NAS must identify the tunnel server to which the call is to be forwarded. You can configure the router to authenticate users and also to select the outgoing tunnel on the basis of the following criteria:

- The user domain name
- The DNIS information in the incoming calls
- Both the domain name and the DNIS information

VPN Tunnel Lookup Based on Domain Name

When a NAS is configured to forward VPN calls on the basis of the user domain name, the user must use a username of the form *username@domain*. The NAS then compares the user domain name to the domain names it is configured to search for. When the NAS finds a match, it forwards the user call to the proper tunnel server.

VPN Tunnel Lookup Based on DNIS Information

When a NAS is configured to forward VPN calls on the basis of the user DNIS information, the NAS identifies the user DNIS information, which is provided on ISDN lines, and then forwards the call to the proper tunnel server.

The ability to select a tunnel on the basis of DNIS information provides additional flexibility to network service providers that offer VPN services and to the corporations that use the services. Instead of having to use only the domain name for tunnel selection, tunnel selection can be based on the dialed number.

With this feature, a corporation—which might have only one domain name—can provide multiple specific phone numbers for users to dial in to the NAS at the service provider POP. The service provider can select the tunnel to the appropriate services or portion of the corporate network on the basis of the dialed number.

VPN Tunnel Lookup Based on Both Domain Name and DNIS Information

When a service provider has multiple AAA servers configured, VPN tunnel authorization searches based on domain name can be time consuming and might cause the client session to time out.

To provide more flexibility, service providers can now configure the NAS to perform tunnel authorization searches by domain name only, by DNIS only, or by both in a specified order.

NAS AAA Tunnel Definition Lookup

Authentication, authorization, and accounting (AAA) tunnel definition lookup allows the NAS to look up tunnel definitions using keywords. Two new Cisco AV pairs are added to support NAS tunnel definition lookup: tunnel type and l2tp-tunnel-password. These AV pairs are configured on the RADIUS server. Descriptions of the values are as follows:

- tunnel type—Indicates that the tunnel type is either L2F or L2TP. This is an optional AV pair and if not defined, reverts to L2F, the default value. If you want to configure an L2TP tunnel, you must use the L2TP AV pair value. This command is case sensitive.
- l2tp-tunnel-password—This value is the secret (password) used for L2TP tunnel authentication and L2TP AV pair hiding. This is an optional AV pair value; however, if it is not defined, the secret will default to the password associated with the local name on the NAS local username-password database. This AV pair is analogous to the **l2tp local secret** command.

For example:

```
request dialin l2tp ip 172.21.9.13 domain hoser.com
l2tp local name dustie
l2tp local secret partner
```

is equivalent to the following RADIUS server configuration:

```
acme.com Password = "cisco"
cisco-avpair = "vpdn: tunnel-id=dustie",
cisco-avpair = "vpdn: tunnel-type=l2tp",
cisco-avpair = "vpdn: l2tp-tunnel-password=partner',
cisco-avpair = "vpdn: ip-addresses=172.21.9.13"
```

Note The password for the domain must be "cisco." This is hard-coded in Cisco IOS software.

L2TP Dial-Out

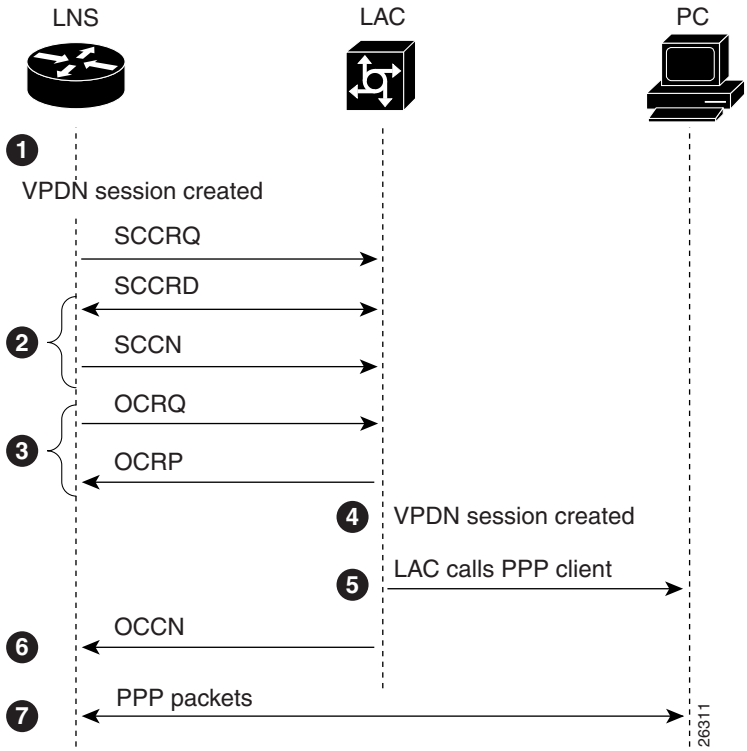
The L2TP dial-out feature enables tunnel servers to tunnel dial-out VPN calls using L2TP as the tunneling protocol. This feature enables a centralized network to efficiently and inexpensively establish a virtual point-to-point connection with any number of remote offices.

Note Cisco routers can carry both dial-in and dial-out calls in the same L2TP tunnels.

L2TP dial-out involves two devices: a tunnel server and a NAS. When the tunnel server wants to perform L2TP dial-out, it negotiates an L2TP tunnel with the NAS. The NAS then places a PPP call to the client(s) that the tunnel server wants to dial out to.

Figure 77 shows a typical L2TP dial-out scenario.

Figure 77 L2TP Dial-Out Process



The following explains the sequence of events described in Figure 77:

1. The tunnel server receives Layer 3 packets, which are to be dialed out, and forwards them to its dialer interface (either a dialer profile or dial-on-demand routing [DDR]).

   The dialer issues a dial call request to the VPN group, and the tunnel server creates a virtual access interface. If the dialer is a dialer profile, this interface becomes a member of the dial pool. If the dialer is DDR, the interface becomes a member of the rotary group.

   The VPN group creates a VPN session for this connection and sets it in the pending state.
2. The tunnel server and NAS establish an L2TP tunnel (unless a tunnel is already open).
3. The tunnel server sends an Outgoing Call ReQuest (OCRQ) packet to the NAS, which checks if it has a dial resource available.

   If the resource is available, the NAS responds to the tunnel server with an Outgoing Call RePly (OCRP) packet. If the resource is not available, the NAS responds with a Call Disconnect Notification (CDN) packet, and the session is terminated.
4. If the NAS has an available resource, it creates a VPN session and sets it in the pending state.
5. The NAS then initiates a call to the PPP client. When the NAS call connects to the PPP client, the NAS binds the call interface to the appropriate VPN session.
6. The NAS sends an Outgoing Call CoNnected (OCCN) packet to the tunnel server. The tunnel server binds the call to the appropriate VPN session and then brings the virtual access interface up.
7. The dialer on the tunnel server and the PPP client can now exchange PPP packets. The NAS acts as a transparent packet forwarder.

If the dialer interface is a DDR and a virtual profile is configured, the PPP endpoint is the tunnel server virtual-access interface, not the dialer. All Layer 3 routes point to this interface instead of the dialer.

Note Large-scale dial-out, Bandwidth Allocation Protocol (BAP), and Dialer Watch are not supported. All configuration must be local on the router.

VPN Configuration Modes Overview

Cisco VPN is configured using the VPN group configuration mode. VPN groups can now support the following:

- One or both of the following tunnel server VPN subgroup configuration modes
  - Accept-dialin
  - Request-dialout
- One or both of the following NAS VPN subgroup configuration modes
  - Request-dialin
  - Accept-dialout
- One of the four VPN subgroup configuration modes

A VPN group can act as either a tunnel server or a NAS, but not both. But individual routers can have both tunnel server VPN groups and NAS VPN groups.

Table 31 list four VPDN group configuration commands that correspond to the configuration modes listed above. These command modes are accessed from VPN group mode; therefore, they are generically referred to as VPN subgroups.

Table 31 New VPN Group Command Modes

Command	Command Mode Prompt	Type of Service
accept-dialin	`router(config-vpdn-acc-in)#`	tunnel server
request-dialout	`router(config-vpdn-req-ou)#`	tunnel server
request-dialin	`router(config-vpdn-req-in)#`	NAS
accept-dialout	`router(config-vpdn-acc-ou)#`	NAS

The keywords and arguments for the previous **accept-dialin** and **request-dialin** VPDN group configuration commands are now independent commands. The previous syntax is still supported, but when you display the configuration, the commands will appear in the new format.

For example, to configure a NAS to request dial-in, you could use the old command, as follows:

```
request-dialin l2tp ip 10.1.2.3 domain jgb.com
```

However when you view the configuration, the keywords and arguments are displayed in the new format with individual commands:

```
request dialin
 protocol l2tp
 domain jgb.com
initiate-to ip 10.1.2.3
```

Similarly, the **accept-dialout** and **request-dialout** commands have subgroup commands that are used to specify information such as the tunneling protocol and dialer resource.

Table 32 lists the new VPN subgroup commands and which command modes they apply to:

Table 32 VPN Subgroup Commands

Command	VPN Subgroups
default	all subgroups
dialer	accept-dialout
dnis	request-dialin
domain	request-dialin
pool-member	request-dialout
protocol	all subgroups
rotary-group	request-dialout
virtual-template	accept-dialin

The other VPN group commands are dependent on which VPN subgroups exist on the VPN group.

Table 33 lists the VPN group commands and which subgroups you need to enable in order for them to be configurable.

Table 33 VPN Group Commands

Command	VPN Subgroups
accept-dialin	tunnel server VPN group[1]
accept-dialout	NAS VPN group[2]
authen before-forward	request-dialin
default	any subgroup
force-local-chap	accept-dialin
initiate-to	request-dialin or request-dialout
lcp renegotiation	accept-dialin
local name	any subgroup
multilink	request-dialin
request-dialin	NAS VPN Group[2]
request-dialout	tunnel server VPN Group[1]
source-ip	any subgroup
terminate-from	accept-dialin or accept-dialout

1. Tunnel server VPN groups can be configured for accept-dialin and/or request-dialout.
2. NAS VPN groups can be configured for accept-dialout and/or request-dialin.

Prerequisites for VPNs

Before configuring a VPN, you must complete the prerequisites described in Table 34. These prerequisites are discussed in the sections that follow.

Table 34 VPN Prerequisites

Prerequisite	Client-Initiated Dial-In	NAS-Initiated Dial-In	Dial-Out
Configuring the LAN Interface	Required	Required	Required
Configuring AAA	Optional	Required	Required
Specifying the IP Address Pool and BOOTP Servers on the Tunnel Server	Required	Required	N/A
Commissioning the T1 Controllers on the NAS	N/A	Required	N/A
Configuring the Serial Channels for Modem Calls on the NAS	N/A	Required	N/A
Configuring the Modems and Asynchronous Lines on the NAS	N/A	Required	N/A
Configuring the Group-Asynchronous Interface on the NAS	N/A	Required	N/A

Table 34 VPN Prerequisites

Prerequisite	Client-Initiated Dial-In	NAS-Initiated Dial-In	Dial-Out
Configuring the Dialer on a NAS	N/A	N/A	Required
Configuring the Dialer on a Tunnel Server	N/A	N/A	Required

Configuring the LAN Interface

To assign an IP address to the interface that will be carrying the VPN traffic and that brings up the interface, use the following commands on both the NAS and the tunnel server beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface interface-type number`	Enters interface configuration mode.
Step 2	`Router(config-if)# ip address ip-address subnet-mask`	Configures the IP address and subnet mask on the interface.
Step 3	`Router(config-if)# no shutdown`	Changes the state of the interface from administratively down to up.

Configuring AAA

To enable AAA, use the following commands on both the NAS and the tunnel server in global configuration mode. If you use RADIUS or TACACS+ for AAA, you also need to point the router to the AAA server using either the **radius-server host** or the **tacacs-server host** command.

Refer to the *Cisco IOS Security Configuration Guide*, Release 12.2, for a complete list of commands and configurable options for security and AAA implementation.

For information on configuring remote AAA servers, refer to the CiscoSecure ACS documentation at: http://www.cisco.com/univercd/cc/td/doc/product/access/acs_soft/index.htm.

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA access control system.
Step 2	`Router(config)# aaa authentication login default {local | radius | tacacs}`	Enables AAA authentication at login and uses the local username database for authentication.[1]
Step 3	`Router(config)# aaa authentication ppp default {local | radius | tacacs}`	Configures the AAA authentication method that is used for PPP and VPN connections.[1]
Step 4	`Router(config)# aaa authorization network default {local | radius | tacacs}`	Configures the AAA authorization method that is used for network-related service requests.[1]
Step 5	`Router(config)# aaa accounting network default start-stop {radius | tacacs}`	(Optional) Enables AAA accounting that sends a stop accounting notice at the end of the requested user process.[1]

	Command	Purpose
Step 6	`Router(config)# vpdn aaa override-server {aaa-server-ip-address | aaa-server-name}`	(Optional) Specifies the AAA servers to be used for VPDN tunnel authorization. If this command is not configured, the default AAA server configured for network authorization is used for VPDN authorization.
Step 7	`Router(config)# vpdn aaa attribute [{nas-ip-address vpdn-nas}| (nas-port vpdn-nas}]`	(Optional) Enables the reporting of AAA attributes from the HGW to the configured RADIUS or TACACS+ AAA server. This command is applicable only on the tunnel server and is disabled by default.
Step 8	`Router(config)# vpdn aaa untagged`	(Optional) Enables the application of untagged attribute values to all attribute sets for VPDN tunnels, unless a value for that attribute is already specified in the attribute set. This command is enabled by default, therefore configuration of this command is required only if the command has been previously disabled.
Step 9	`Router(config)# radius-server host ip-address [auth-port number] [acct-port number]`	Specifies the RADIUS server IP address and optionally the ports to be used for authentication and accounting requests.
	`Router(config)# radius-server key cisco`	Sets the authentication key and encryption key for all RADIUS communication. **Note** The RADIUS key must be "cisco." This is hard-coded in Cisco IOS software.
	or `Router(config)# tacacs-server host ip-address [port integer] [key string]`	Specifies the TACACS+ server IP address and optionally the port to be used, and an authentication and encryption key.

1. If you specify more than one method, AAA will query the servers or databases in the order that they are entered.

Specifying the IP Address Pool and BOOTP Servers on the Tunnel Server

To specify the IP addresses and the BOOTP servers that will be assigned to VPN clients, use the following commands on the tunnel server in global configuration mode.

The IP address pool is the addresses that the tunnel server assigns to clients. You must configure an IP address pool. You can also provide BOOTP servers. Domain Name System (DNS) servers translate host names to IP addresses. WINS servers, which are specified using the **async-bootp nbns-server** command, provide dynamic NetBIOS names that Windows devices use to communicate without IP addresses.

	Command	Purpose
Step 1	`HGW(config)# ip local pool default first-ip-address last-ip-address`	Configures the default local pool of IP address that will be used by clients.
Step 2	`HGW(config)# async-bootp dns-server ip-address1 [additional-ip-address]`	(Optional) Returns the configured addresses of DNS in response to BOOTP requests.
Step 3	`HGW(config)# async-bootp nbns-server ip-address1 [additional-ip-address]`	(Optional) Returns the configured addresses of Windows NT servers in response to BOOTP requests.

Commissioning the T1 Controllers on the NAS

To define the ISDN switch type and commission the T1 controllers to allow modem calls to come into the NAS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`NAS(config)# isdn switch-type switch-type`	Enters the telco switch type. An ISDN switch type that is specified in global configuration mode is automatically propagated into the individual serial interfaces (for example, serial interface 0:23, 1:23, 2:23, and 3:23).
Step 2	`NAS(config)# controller t1 0`	Accesses controller configuration mode for the first T1 controller, which is number 0. The controller ports are numbered 0 through 3 on the quad T1/PRI card.
Step 3	`NAS(config-controller)# framing framing-type`	Enters the T1 framing type.
Step 4	`NAS(config-controller)# linecode linecode`	Enters the T1 line-code type.

	Command	Purpose
Step 5	`NAS(config-controller)# clock source line primary`	Configures the access server to get its primary clocking from the T1 line assigned to controller 0. Line clocking comes from the remote switch.
Step 6	`NAS(config-controller)# pri-group timeslots` *range*	Assigns the T1 time slots as ISDN PRI channels. After you enter this command, a D-channel serial interface is instantly created (for example, S0:23), along with individual B-channel serial interfaces (S0:0, S0:1, and so on). The D-channel interface functions like a dialer for the B channels using the controller. If this was an E1 interface, the PRI group range would be 1 to 31. The D-channel serial interfaces would be S0:15, S1:15, S2:15, and S3:15.

Configuring the Serial Channels for Modem Calls on the NAS

To configure the D channels (the signaling channels) to allow incoming voice calls to be routed to the integrated MICA technologies modems and to control the behavior of the individual B channels, use the following commands on the NAS beginning in global configuration mode:

	Command	Purpose
Step 1	`NAS(config)# interface serial 0:23`	Accesses configuration mode for the D-channel serial interface that corresponds to controller T1 0. The behavior of serial 0:0 through serial 0:22 is controlled by the configuration instructions provided for serial 0:23. This concept is also true for the other remaining D-channel configurations.
Step 2	`NAS(config-if)# isdn incoming-voice modem`	Enables analog modem voice calls that come in through the B channels to be connected to the integrated modems.
Step 3	`NAS(config-if)# exit`	Returns to global configuration mode.

Step 4	`NAS(config)# interface serial 1:23`
	`NAS(config-if)# isdn incoming-voice modem`
	`NAS(config-if)# exit`
	`NAS(config)# interface serial 2:23`
	`NAS(config-if)# isdn incoming-voice modem`
	`NAS(config-if)# exit`
	`NAS(config)# interface serial 3:23`
	`NAS(config-if)# isdn incoming-voice modem`
	`NAS(config-if)# exit`
		Configures the three remaining D channels with the same ISDN incoming-voice modem setting.

Configuring the Modems and Asynchronous Lines on the NAS

To define a range of modem lines and to enable PPP clients to dial in, bypass the EXEC facility, and automatically start PPP, use the following commands on the NAS beginning in global configuration mode.

Configure the modems and lines after the ISDN channels are operational. Each modem corresponds with a dedicated asynchronous line inside the NAS. The modem speed of 115200 bps and hardware flow control are default values for integrated modems.

	Command	Purpose
Step 1	`NAS(config)# line line-number [ending-line-number]`	Enters the modem line or range of modem lines (by entering an *ending-line-number*) that you want to configure.
Step 2	`NAS(config-line)# autoselect ppp`	Enables PPP clients to dial in, bypass the EXEC facility, and automatically start PPP on the lines.
Step 3	`NAS(config-line)# autoselect during-login`	Displays the username:password prompt as the modems connect. **Note** These two autoselect commands enable EXEC (shell) and PPP services on the same lines.
Step 4	`NAS(config-line)# modem inout`	Supports incoming and outgoing modem calls.

Configuring the Group-Asynchronous Interface on the NAS

To create a group-asynchronous interface and project protocol characteristics to the asynchronous interfaces, use the following commands on the NAS beginning in global configuration mode.

The group-async interface is a template that controls the configuration of the specified asynchronous interfaces inside the NAS. Asynchronous interfaces are lines running in PPP mode. An asynchronous interface uses the same number as its corresponding line. Configuring all the asynchronous interfaces as an asynchronous group saves you time by reducing the number of configuration steps.

	Command	Purpose
Step 1	`NAS(config)# interface group-async number`	Creates the group-asynchronous interface.
Step 2	`NAS(config-if)# ip unnumbered interface-type number`	Uses the IP address defined on the specified interface.
Step 3	`NAS(config-if)# encapsulation ppp`	Enables PPP.
Step 4	`NAS(config-if)# async mode interactive`	Configures interactive mode on the asynchronous interfaces. Interactive mode means that clients can dial in to the NAS and get a router prompt or PPP session. Dedicated mode means that only PPP sessions can be established on the NAS. Clients cannot dial in and get an EXEC (shell) session.

	Command	Purpose
Step 5	`NAS(config-if)# ppp authentication {chap | pap | chap pap | pap chap}`	Configures the authentication to be used on the interface during LCP negotiation. When both authentication methods are specified, the NAS first authenticates with the first method entered. If the first method is rejected by the client, the second authentication method is used.
Step 6	`NAS(config-if)# group-range range`	Specifies the range of asynchronous interfaces to include in the group, which is usually equal to the number of modems in the access server.

Configuring the Dialer on a NAS

To configure the dialer on a NAS for L2TP dial-out, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`NAS(config)# interface dialer number`	Defines a dialer rotary group.
Step 2	`NAS(config-if)# ip unnumbered interface-type number`	Configures the dialer to use the interface IP address.
Step 3	`NAS(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`NAS(config-if)# dialer in-band`	Enables DDR on the dialer.
Step 5	`NAS(config-if)# dialer aaa`	Enables the dialer to use the AAA server to locate profiles for dialing information.
Step 6	`NAS(config-if)# dialer-group group-number`	Assigns the dialer to the specified dialer group.
Step 7	`NAS(config-if)# ppp authentication chap`	Specifies that CHAP authentication will be used.

Configuring the Dialer on a Tunnel Server

To configure the dialer on an a tunnel server for L2TP dial-out, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`LNS(config)# interface dialer number`	Defines a dialer rotary group.
Step 2	`LNS(config-if)# ip address ip-address subnet-mask`	Specifies an IP address for the group.
Step 3	`LNS(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`LNS(config-if)# dialer remote-name peer-name`	Specifies the name used to authenticate the remote router that is being dialed.
Step 5	`LNS(config-if)# dialer string dialer-number`	Specifies the number that is dialed.
Step 6	`LNS(config-if)# dialer vpdn`	Enables dial-out.
Step 7	`LNS(config-if)# dialer pool pool-number`	Specifies the dialer pool.

	Command	Purpose
Step 8	`LNS(config-if)# dialer-group group-number`	Assigns the dialer to the specified dialer group.
Step 9	`LNS(config-if)# ppp authentication chap`	Specifies that CHAP authentication will be used.

How to Configure a VPN

Configuration for both dial-in and dial-out VPNs is described in the following sections:

- Enabling a VPN
- Configuring VPN Tunnel Authentication Using the Host Name or Local Name
- Configuring VPN Tunnel Authentication Using the L2TP Tunnel Password
- Configuring Client-Initiated Dial-In VPN
- Configuring NAS-Initiated Dial-In VPN
- Configuring Dial-Out VPN
- Configuring Advanced VPN Features

See the section "Configuration Examples for VPN" later in this chapter for examples of how you can implement VPN in your network.

Enabling a VPN

To enable a VPN tunnel, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vpdn[1] enable`	Enables VPN.

1. The Cisco IOS command syntax uses the more specific term VPDN (virtual private dialup network) instead of VPN.

To disable a VPN tunnel, use the **clear vpdn tunnel** command in EXEC mode. The **no vpdn enable** command does not automatically disable a VPN tunnel.

Configuring VPN Tunnel Authentication Configuration

VPN tunnel authentication enables routers to authenticate the other tunnel endpoint before establishing a VPN tunnel. It is required for L2F tunnels and optional for L2TP tunnels.

Disabling VPN Tunnel Authentication for L2TP Tunnels

To disable VPN tunnel authentication for L2TP tunnels, use the following commands beginning in global configuration mode:

Command	Purpose
`ISP_NAS(config)# vpdn-group group` `ISP_NAS(config-vpdn)# no l2tp tunnel authentication`	Disables VPN tunnel authentication for the specified VPN group. The VPN group will not challenge any router that attempts to open an L2TP tunnel.

Note Before you can configure any **l2tp** VPN group command, you must specify L2TP as the protocol for a VPN subgroup within the VPN group. For more information, see the "Configuring NAS-Initiated Dial-In VPN" and "Configuring Dial-Out VPN" sections later in this chapter.

VPN tunnel authentication can be performed in the following ways:

- Using local AAA on both the NAS and the tunnel server
- Using RADIUS on the NAS and local AAA on the tunnel server
- Using TACACS+ on the NAS and local AAA on the tunnel server

This section discusses local tunnel authentication. For information on RADIUS and TACACS+, refer to the "NAS AAA Tunnel Definition Lookup" section earlier in this chapter and the *Cisco IOS Security Configuration Guide*, Release 12.2.

VPN tunnel authentication requires that a single shared secret—called the *tunnel secret*—be configured on both the NAS and tunnel server. There are two methods for configuring the tunnel secret:

- Configuring VPN Tunnel Authentication Using the Host Name or Local Name

  The tunnel secret is configured as a password by using the **username** command.
- Configuring VPN Tunnel Authentication Using the L2TP Tunnel Password

  The tunnel secret is configured by using the **l2tp tunnel password** command.

Configuring VPN Tunnel Authentication Using the Host Name or Local Name

To configure VPN tunnel authentication using the **hostname** or **local name** commands, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`ISP_NAS(config)# hostname host-name`	Configures the router host name. By default, the router uses the host name as the tunnel name in VPN tunnel authentication.

	or
	`ISP_NAS(config)# vpdn-group group`
	`ISP_NAS(config-vpdn)# local name tunnel-name`
		or
		(Optional) Configures the local name for the VPN group. When negotiating VPN tunnel authentication for this VPN group, the router will use the local name as the tunnel name.

Step 2	`ISP_NAS(config)# username tunnel-name password tunnel-secret`	Configures the other router's tunnel name and the tunnel secret as a user name and password combination. **Note** The tunnel secret must be the same on both routers. Each router must have the other router's tunnel name (specified by either the **hostname** or **local name** command) configured as a username with the tunnel secret as the password.

Configuring VPN Tunnel Authentication Using the L2TP Tunnel Password

To configure VPN tunnel authentication using the **l2tp tunnel password** command, use the following commands beginning in global configuration:

	Command	Purpose
Step 1	`ISP_NAS(config)# vpdn-group group` `ISP_NAS(config-vpdn)# l2tp tunnel password tunnel-secret`	Configures the tunnel secret that will be used for VPN tunnel authentication for this VPN group and enters VPDN configuration mode.

Step 2	`ISP_NAS(config-vpdn)# local name tunnel-name`
	`ISP_NAS(config-vpdn)# exit`
	
	`ISP_NAS(config)# username tunnel-name password tunnel-secret`
		(Optional) Configures the tunnel name of the router.
		
		(Optional) Configures the other router's tunnel name and the tunnel secret as a user name.
		If the other router uses the **l2tp tunnel password** command to configure the tunnel secret, these commands are not necessary.
		Note The tunnel secret must be the same on both routers.

For sample VPN tunnel authentication configurations, see the "VPN Tunnel Authentication Examples" section later in this chapter.

Configuring Client-Initiated Dial-In VPN

For client-initiated dial-in VPNs, complete the following tasks:

- Configuring a Tunnel Server to Accept Dial-In (Required)
- Configuring MPPE on the ISA Card (Optional)
- Tuning PPTP (Optional)

When configuring PPTP and MPPE, you should consider the following restrictions:

- Only Cisco Express Forwarding (CEF) and process switching are supported. Regular fast switching is not supported.
- PPTP does not support multilink.
- VPDN multihop is not supported.
- Because all PPTP signaling is over TCP, TCP configurations will affect PPTP performance in large-scale environments.
- MPPE is not supported with TACACS.
- MPPE is supported with RADIUS in Cisco IOS Releases 12.0(7)XE1 and later releases.
- Windows clients must use MS-CHAP authentication in order for MPPE to work.
- If you are performing mutual authentication with MS-CHAP and MPPE, both sides of the tunnel must use the same password.
- To use MPPE with AAA, you must use a RADIUS server that supports the Microsoft Vendor specific attribute for MPPE-KEYS. CiscoSecure NT supports MPPE beginning with release 2.6. CiscoSecure UNIX does not support MPPE.

Configuring a Tunnel Server to Accept PPTP Tunnels

To configure a tunnel to accept tunneled PPP connections from a client, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`PNS(config)# vpdn-group 1`	Creates vpdn group 1.
Step 2	`PNS(config-vpdn)# accept-dialin`	Enables the tunnel server to accept dial-in requests.
Step 3	`PNS(config-vpdn-acc-in)# protocol pptp`	Specifies that the tunneling protocol will be PPTP.
Step 4	`PNS(config-vpdn-acc-in)# virtual-template` *template-number*	Specifies the number of the virtual template that will be used to clone the virtual-access interface.
Step 5	`PNS(config-vpdn-acc-in)# exit`	Exit to higher command mode.
Step 6	`PNS(config-vpdn)# local name` *localname*	(Optional) Specifies that the tunnel server will identify itself with this local name. If no local name is specified, the tunnel server will identify itself with its host name.

Configuring MPPE on the ISA Card

To offload MPPE encryption from the tunnel server processor to the ISA card, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`PNS(config)# controller isa slot/port`	Enters controller configuration mode on the ISA card.
Step 2	`PNS(config-controller)# encryption mppe`	Enables MPPE encryption

Tuning PPTP

To tune PPTP, use one or more of the following commands in VPDN configuration mode:

Command	Purpose
`PNS(config-vpdn)# pptp flow-control receive-window packets`	Specifies how many packets the client can send before it must wait for the acknowledgment from the tunnel server.
`PNS(config-vpdn)# pptp flow-control static-rtt milliseconds`	Specifies the timeout interval of the tunnel server between sending a packet to the client and receiving a response.
`PNS(config-vpdn)# pptp tunnel echo seconds`	Specifies the period of idle time on the tunnel that will trigger an echo message from the tunnel server to the client.

Configuring NAS-Initiated Dial-In VPN

The following tasks must be completed for NAS-initiated dial-in VPNs:

- Configuring a NAS to Request Dial-In (Required)
- Configuring a Tunnel Server to Accept Dial-In (Required)
- Creating the Virtual Template on the Network Server (Required)

Configuring a NAS to Request Dial-In

The NAS is a device that is typically (although not always) located at a service provider POP; initial configuration and ongoing management are done by the service provider.

To configure a NAS to accept PPP calls and tunnel them to a tunnel server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`NAS(config)# vpdn-group 1`	Creates VPN group 1.
Step 2	`NAS(config-vpdn)# request-dialin`	Enables the NAS to request L2F or L2TP dial-in requests.
Step 3	`NAS(config-vpdn-req-in)# protocol [l2f | l2tp | any]`	Specifies which tunneling protocol is to be used.

	Command	Purpose

Step 4	`NAS(config-vpdn-req-in)# domain domain-name`
	
	or
	
	`NAS(config-vpdn-req-in)# dnis dnis-number`
		Specifies the domain name of the users that are to be tunneled.
		
		Specifies the DNIS number of users that are to be tunneled.
		
		You can configure multiple domain names and/or DNIS numbers for an individual request-dialin subgroup.

Step 5	`NAS(config-vpdn-req-in)# exit` `NAS(config-vpdn)# initiate-to ip ip-address`	Specifies the IP address that the NAS will establish the tunnel with. This is the IP address of the tunnel server.
Step 6	`NAS(config-vpdn)# vpdn search-order {domain | dnis | domain dnis | dnis domain}`	(Optional) Specifies the method that is used to determine if a dial-in call should be tunneled. If both keywords are entered, the NAS will search the criteria in the order they are entered.

Configuring a Tunnel Server to Accept Dial-In

To configure a tunnel server to accept tunneled PPP connections from a NAS, use the following commands beginning in global configuration mode.

The tunnel server is the termination point for a VPN tunnel. The tunnel server initiates outgoing calls to and receives incoming calls from the NAS.

	Command	Purpose
Step 1	`LNS(config)# vpdn-group 1`	Creates VPN group 1.
Step 2	`LNS(config-vpdn)# accept-dialin`	Enables the tunnel server to accept dial-in requests.
Step 3	`LNS(config-vpdn-acc-in)# protocol [l2f | l2tp | any]`	Specifies which tunneling protocol is to be used.
Step 4	`LNS(config-vpdn-acc-in)# virtual-template number`	Specifies the number of the virtual template that will be used to clone the virtual access interface.
Step 5	`LNS(config-vpdn-acc-in)# exit` `LNS(config-vpdn)# terminate-from hostname hostname`	Accepts tunnels that have this host name configured as a local name.

See the section "Tunnel Server Comprehensive Dial-in Configuration Example" later in this chapter for a configuration example.

Creating the Virtual Template on the Network Server

At this point, you can configure the virtual template interface with configuration parameters you want applied to virtual access interfaces. A virtual template interface is a logical entity configured for a serial interface. The virtual template interface is not tied to any physical interface and is applied dynamically, as needed. Virtual access interfaces are *cloned* from a virtual template interface, used on demand, and then freed when no longer needed.

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`HGW(config)# interface virtual-template number`	Create the virtual template that is used to clone virtual access interfaces.
Step 2	`HGW(config-if)# ip unnumbered interface-type number`	Specifies that the virtual access interfaces use the specified interface IP address.
Step 3	`HGW(config-if)# ppp authentication {chap | pap | chap pap | pap chap}`	Enables CHAP authentication using the local username database.
Step 4	`HGW(config-if)# peer default ip address pool pool`	Returns an IP address from the default pool to the client.
Step 5	`HGW(config-if)# encapsulation ppp`	Enables PPP encapsulation.

Optionally, you can configure other commands for the virtual template interface. For more information about configuring virtual template interfaces, refer to the "Configuring Virtual Template Interfaces" chapter in this publication.

Configuring Dial-Out VPN

The following tasks must be completed for dial-out VPNs:

- Configuring a Tunnel Server to Request Dial-Out (Required)
- Configuring a NAS to Accept Dial-Out (Required)

Configuring a Tunnel Server to Request Dial-Out

To configure a tunnel server to request dial-out tunneled PPP connections to a NAS, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`LNS(config)# vpdn-group 1`	Creates VPN group 1.
Step 2	`LNS(config-vpdn)# request-dialout`	Enables the tunnel server to send L2TP dial-out requests.
Step 3	`LNS(config-vpdn-req-ou)# protocol l2tp`	Specifies L2TP as the tunneling protocol. **Note** L2TP is the only protocol that supports dial-out.

Step 4	`LNS(config-vpdn-req-ou)# pool-member pool-number`
	or
	`LNS(config-vpdn-req-ou)# rotary-group group-number`
		Specifies the dialer profile pool that will be used to dial out.
		Specifies the dialer rotary group that will be used to dial out.
		You can configure only one dialer profile pool or dialer rotary group. Attempting to configure a second dialer resource will remove the first from the configuration.

	Command	Purpose
Step 5	`LNS(config-vpdn-req-ou)# exit` `LNS(config-vpdn)# initiate-to ip ip-address`	Specifies the IP address that will be dialed out. This is the IP address of the NAS.
Step 6	`LNS(config-vpdn)# local name hostname`	Specifies that the L2TP tunnel will identify itself with this host name.

Configuring a NAS to Accept Dial-Out

To configure a NAS to accept tunneled dial-out connections from a tunnel server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`NAS(config)# vpdn-group 1`	Creates VPN group 1.
Step 2	`NAS(config-vpdn)# accept-dialout`	Enables the NAS to accept L2TP dial-out requests.
Step 3	`NAS(config-vpdn-acc-ou)# protocol l2tp`	Specifies L2TP as the tunneling protocol. **Note** L2TP is the only protocol that supports dial-out.
Step 4	`NAS(config-vpdn-acc-ou)# dialer dialer-interface`	Specifies the dialer that is used to dial out to the client.
Step 5	`NAS(config-vpdn-acc-ou)# exit` `NAS(config-vpdn)# terminate-from hostname hostname`	Accepts L2TP tunnels that have this host name configured as a local name.

Configuring Advanced VPN Features

The following optional tasks provide advanced VPN features:

- Configuring Advanced Remote AAA Features
- Configuring Per-User VPN
- Configuring Preservation of IP ToS Field
- Shutting Down a VPN Tunnel
- Limiting the Number of Allowed Simultaneous VPN Sessions
- Enabling Soft Shutdown of VPN Tunnels
- Configuring Event Logging
- Setting the History Table Size

Configuring Advanced Remote AAA Features

This section describes the following two advanced remote AAA features for VPNs:

- Tunnel Server Load Balancing on the NAS AAA Server
- DNS Name Support

Tunnel Server Load Balancing on the NAS AAA Server

NAS AAA servers can forward users of the same domain name or DNIS to more than one tunnel server. The NAS AAA server can be configured to balance the load of calls equally among the tunnel servers, or it can designate different priority levels to the tunnel servers.

To configure load balancing on a NAS RADIUS server, configure multiple IP addresses in the vpdn:ip-addresses attribute value (AV) pair. The IP addresses can be separated by either spaces or by commas. The following example shows a profile that will equally balance the load between three tunnel servers.

```
user = terrapin.com{
profile_id = 29
profile_cycle = 7
radius=Cisco {
check_items= {
2=cisco
}
reply_attributes= {
9,1="vpdn:l2tp-tunnel-password=cisco123"
9,1="vpdn:tunnel-type=l2tp"
9,1="vpdn:ip-addresses=172.16.171.11 172.16.171.12 172.16.171.13"
9,1="vpdn:tunnel-id=tunnel"
}
}
}
```

To specify different priorities for the tunnel servers, separate the IP addresses with a slash. The following AV pair instructs the RADIUS server to equally balance calls between 172.16.171.11 and 172.16.171.12. If both of those tunnel servers are unavailable, the RADIUS server will tunnel calls to 172.16.171.13.

```
9,1="vpdn:ip-addresses=172.16.171.11 172.16.171.12/172.16.171.13"
```

DNS Name Support

NAS AAA servers can resolve DNS names and translate them into IP addresses. The server will first look up the name in its name cache. If the name is not in the name cache, the server will resolve the name by using a DNS server. The following AV pair instructs the RADIUS server to resolve the DNS name "terrapin" and tunnel calls to the appropriate IP addresses:

```
9,1="vpdn:ip-addresses=terrapin"
```

For detailed information about remote AAA configuration, refer to the CiscoSecure ACS documentation at http://www.cisco.com/univercd/cc/td/doc/product/access/acs_soft/index.htm.

Configuring Per-User VPN

In a VPN that uses remote AAA, when a user dials in, the access server that receives the call forwards information about the user to its remote AAA server. With basic VPN, the access server sends only the user domain name (when performing authentication based on domain name) or the telephone number the user dialed in from (when performing authentication based on DNIS).

Per-user VPN configuration sends the entire structured username to the AAA server the first time the router contacts the AAA server. This enables Cisco IOS software to customize tunnel attributes for individual users who use a common domain name or DNIS.

Without VPN per-user configuration, Cisco IOS software sends only the domain name or DNIS to determine VPN tunnel attribute information. Then, if no VPN tunnel attributes are returned, Cisco IOS software sends the entire username string.

Note Per-user VPN configuration supports only RADIUS as the AAA protocol.

To configure per-user VPN, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group group-number`	Enters VPN group configuration mode.
Step 2	`Router(config-vpdn)# authen before-forward`	Specifies that the entire structured username be sent to the AAA server the first time the router contacts the AAA server.

Configuring Preservation of IP ToS Field

When L2TP data packets are created, they have a type of service (ToS) field of zero, which indicates normal service. This ignores the ToS field of the encapsulated IP packets that are being tunneled.

To preserve quality of service (QoS) for tunneled packets by copying the ToS field of the IP packets' onto the L2TP data packets when they are created at the tunnel server virtual access interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`LNS(config)# vpdn-group 1`	Creates VPN group 1.

Step 2	`LNS(config-vpdn)# accept-dialin`
	or
	`LNS(config-vpdn)# request-dialout`
		Enables the tunnel server to accept dial-in requests.
		
		Enables the tunnel server to send L2TP dial-out requests.

Step 3	`LNS(config-vpdn-acc-in)# protocol l2tp`
	or
	`LNS(config-vpdn-req-ou)# protocol l2tp`
		Specifies L2TP as the tunneling protocol.
		Note L2TP is the only protocol that supports dial-out and IP ToS preservation.

Step 4	`LNS(config-vpdn-req-ou)# exit`	Returns to VPDN group configuration mode.
Step 5	`LNS(config-vpdn)# ip tos reflect`	Preserves the ToS field of the encapsulated IP packets.

Note The tunneled link must carry IP for the ToS field to be preserved. The encapsulated payload of Multilink PPP (MLP) connections is not IP, therefore this task has no effect when MLP is tunneled.

Note Proxy PPP dial-in is not supported.

Shutting Down a VPN Tunnel

To shut down a VPN tunnel, use the following command in privileged EXEC mode:

Command	Purpose
`Router# clear vpdn tunnel {l2f nas-name hgw-name | l2tp [remote-name] [local-name]}`	Shuts down a specific tunnel and all the sessions within the tunnel.

Limiting the Number of Allowed Simultaneous VPN Sessions

To set a limit for the maximum number of allowed simultaneous VPN sessions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vpdn session-limit sessions`	Limits the number of simultaneous VPN sessions on the router to the number specified with the *sessions* argument.

To verify that the **vpdn session-limit** command is working properly, perform the following steps:

Note If you use a Telnet session to connect to the NAS, enable the **terminal monitor** command, which ensures that your EXEC session is receiving the logging and debug output from the NAS.

Step 1 Enter the **vpdn session-limit 1** global configuration command on either the NAS or tunnel server.

Step 2 Establish a VPN session by dialing in to the NAS using an allowed username and password.

Step 3 Attempt to establish another VPN session by dialing in to the NAS using another allowed username and password.

Step 4 A Syslog message similar to the following should appear on the console of the router:

```
00:11:17:%VPDN-6-MAX_SESS_EXCD:L2F HGW great_went has exceeded configured local
session-limit and rejected user wilson@soam.com
```

Step 5 Enter the **show vpdn history failure** command on the router. If you see output similar to the following, the session limit was successful:

```
User:wilson@soam.com
 NAS:cliford_ball, IP address = 172.25.52.8, CLID = 2
 Gateway:great_went, IP address = 172.25.52.7, CLID = 13
 Log time:00:04:21, Error repeat count:1
 Failure type:Exceeded configured VPDN maximum session limit.
 Failure reason:
```

Enabling Soft Shutdown of VPN Tunnels

To prevent new sessions from being established on a VPN tunnel without disturbing the service of existing sessions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# vpdn softshut`[1]	Prevents new sessions from being established on a VPN tunnel without disturbing existing sessions.

1. When the **vpdn softshut** command is enabled, Multichassis Multilink PPP (MMP) L2F tunnels can still be created and established.

When the **vpdn softshut** command is enabled on a NAS, the potential session will be authorized before it is refused. This authorization ensures that accurate accounting records can be kept.

When the **vpdn softshut** command is enabled on a tunnel server, the reason for the session refusal will be returned to the NAS. This information is recorded in the VPN history failure table.

To verify that the **vpdn softshut** command is working properly, perform the following steps:

Step 1 Establish a VPN session by dialing in to the NAS using an allowed username and password.

Step 2 Enter the **vpdn softshut** global configuration command on either the NAS or the tunnel server.

Step 3 Verify that the original session is still active by entering the **show vpdn** command:

```
ENT_HGW# show vpdn

% No active L2TP tunnels

L2F Tunnel and Session

 NAS CLID HGW CLID NAS Name        HGW Name        State
 36       1        cliford_ball    great_went      open
                   172.25.52.8     172.25.52.7

 CLID   MID    Username                   Intf   State
 36     1      mockingbird@gamehendge.com Vi1    open
```

Step 4 Attempt to establish another VPN session by dialing in to the NAS using another allowed username and password.

Step 5 A Syslog message similar to the following should appear on the console of the soft shutdown router:

```
00:11:17:%VPDN-6-SOFTSHUT:L2F HGW great_went has turned on softshut and rejected user
wilson@soam.com
```

Step 6 Enter the **show vpdn history failure** command on the soft shutdown router. If you see output similar to the following, the soft shutdown was successful:

```
User:wilson@soam.com
NAS:cliford_ball, IP address = 172.25.52.8, CLID = 2
Gateway:great_went, IP address = 172.25.52.7, CLID = 13
Log time:00:04:21, Error repeat count:1
Failure type:VPDN softshut has been activated.
Failure reason:
```

Configuring Event Logging

The Syslog mechanism provides generic and failure event logging. Generic logging is a mixture of type error, warning, notification, and information logging for VPN. Logging can be done locally or at a remote tunnel destination. Both generic and failure event logging is enabled by default; therefore, if you wish to disable VPN failure events you must specifically configure the router or access server to do so. In order to disable the router to log VPN generic or history events, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# vpdn logging [local | remote]`	Enables generic event logging, locally or at a remote endpoint.
`Router(config)# vpdn history failure`	Enables the logging of failure events to the failure history table. **Note** By default, VPN failure history logging is enabled.

Setting the History Table Size

You may set the failure history table to a specific number of entries based on the amount of data you wish to track. To set the failure history table, use the following commands in global configuration mode:

Command	Purpose
`Router(config)# vpdn history failure table-size` *entries*	(Optional) Sets the failure history table depth.

Verifying VPN Sessions

The following sections detail the procedures used for verifying VPN sessions:

- Verifying a Client-Initiated VPN
- Verifying a NAS-Initiated VPN

Verifying a Client-Initiated VPN

To verify that a PPTP network functions properly, complete the following verification steps:

Step 1 From the client, dial in to the ISP and establish a PPP session.

Step 2 From the client, dial in to the tunnel server.

Step 3 From the client, ping the tunnel server. From the client desktop:

a. Click **Start**.

b. Select **Run**.

c. Enter **ping** *tunnel-server-ip-address.*

d. Click **OK**.

e. Look at the terminal screen and verify that the tunnel server is sending ping reply packets to the client.

Step 4 From the tunnel server, enter the **show vpdn** command and verify that the client has established a PPTP session.

```
PNS# show vpdn

% No active L2TP tunnels

% No active L2F tunnels

PPTP Tunnel and Session Information (Total tunnels=1 sessions=1)

LocID RemID Remote Name     State    Remote Address  Port  Sessions
13    13    10.1.2.41       estabd   10.1.2.41       1136  1

LocID RemID TunID Intf    Username      State   Last Chg
13    0     13    Vi3                   estabd  000030
```

Step 5 For more detailed information, enter the **show vpdn session all** or **show vpdn session window** commands. The last line of output from the **show vpdn session all** command indicates the current status of the flow control alarm.

```
PNS# show vpdn session all

% No active L2TP tunnels

% No active L2F tunnels

PPTP Session Information (Total tunnels=1 sessions=1)

Call id 13 is up on tunnel id 13
Remote tunnel name is 10.1.2.41
 Internet Address is 10.1.2.41
 Session username is unknown, state is estabd
 Time since change 000106, interface Vi3
 Remote call id is 0
 10 packets sent, 10 received, 332 bytes sent, 448 received
 Ss 11, Sr 10, Remote Nr 10, peer RWS 16
 0 out of order packets
 Flow alarm is clear.
```

The last line of output from the **show vpdn session window** command indicates the current status of the flow control alarm (under the heading “Congestion”) and the number of flow control alarms that have gone off during the session (under the heading “Alarms”).

```
PNS# show vpdn session window

% No active L2TP tunnels
% No active L2F tunnels
PPTP Session Information (Total tunnels=1 sessions=1)

LocID RemID TunID ZLB-tx  ZLB-rx  Congestion Alarms   Peer-RWS
13    0     13    0       1       clear      0        16
```

Step 6 For information on the virtual-access interface, enter the **show ppp mppe virtual-access** *number* command:

```
PNS# show ppp mppe virtual-access3

Interface Virtual-Access3 (current connection)
  Hardware (ISA5/1, flow_id=13) encryption, 40 bit encryption, Stateless mode
  packets encrypted = 0        packets decrypted  = 1
```

```
  sent CCP resets    = 0         receive CCP resets = 0
  next tx coherency = 0         next rx coherency  = 0
  tx key changes     = 0         rx key changes     = 0
  rx pkt dropped     = 0         rx out of order pkt= 0
  rx missed packets = 0
```

To update the key change information, reissue the **show ppp mppe virtual-access3** command.

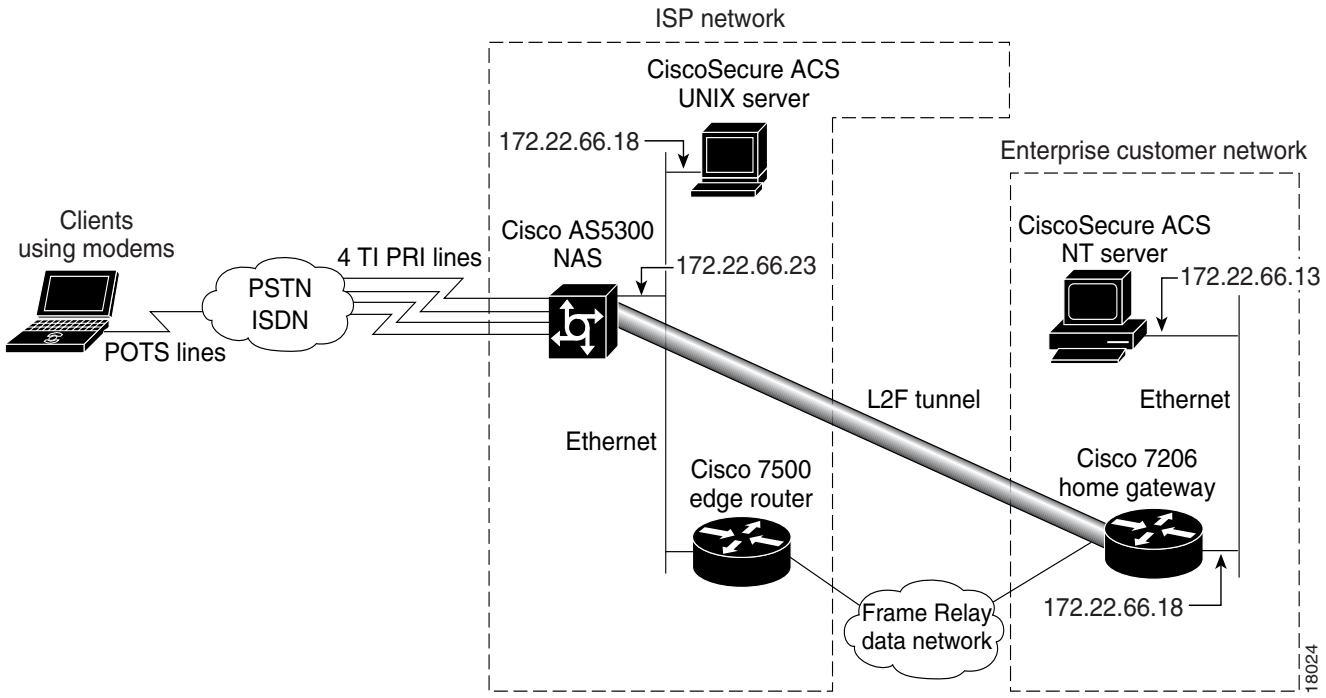
```
PNS# show ppp mppe virtual-access3

Interface Virtual-Access3 (current connection)
  Hardware (ISA5/1, flow_id=13) encryption, 40 bit encryption, Stateless mode
  packets encrypted = 0         packets decrypted  = 1
  sent CCP resets    = 0         receive CCP resets = 0
  next tx coherency = 0         next rx coherency  = 0
  tx key changes     = 0         rx key changes     = 1
  rx pkt dropped     = 0         rx out of order pkt= 0
  rx missed packets = 0
```

Verifying a NAS-Initiated VPN

This section describes how to verify that an L2F dial-in scenario functions as shown in Figure 78. To verify connectivity, complete the following verification steps:

- Step 1: Dialing In to the NAS
- Step 2: Pinging the Tunnel Server
- Step 3: Displaying Active Call Statistics on the Tunnel Server
- Step 4: Pinging the Client
- Step 5: Verifying That the Virtual-Access Interface Is Up and That LCP Is Open
- Step 6: Viewing Active L2F Tunnel Statistics

Figure 78 L2F Dial-In Topology Using Remote AAA



Step 1 From the client, dial in to the NAS by using the PRI telephone number assigned to the NAS T1 trunks. Sometimes this telephone number is called the hunt group number.

As the call comes in to the NAS, a LINK-3-UPDOWN message automatically appears on the NAS terminal screen. In the following example, the call comes in to the NAS on asynchronous interface 14. The asynchronous interface is up.

```
*Jan  1 21:22:18.410: %LINK-3-UPDOWN: Interface Async14, changed state to up
```

Note No **debug** commands are turned on to display this log message. Start troubleshooting the NAS if you do not see this message 30 seconds after the client first sends the call.

Step 2 From the client, ping the tunnel server. From the client Windows 95 desktop, perform the following steps:

a. Click **Start**.

b. Select **Run**.

c. Enter the **ping** *ip-address* command, where the IP address is the tunnel server address.

d. Click **OK**.

e. Look at the terminal screen and verify that the tunnel server is sending ping reply packets to the client.

Step 3 From the tunnel server, enter the **show caller** command and the **show caller user** *name* command to verify that the client received an IP address. The following example shows that Jeremy is using interface virtual-access 1 and IP address 172.30.2.1. The network administrator jane-admin is using console 0.

```
ENT_HGW# show caller
  Line         User               Service       Active
  con 0        jane-admin         TTY           00:00:25
  Vi1          jeremy@hgw.com     PPP   L2F     00:01:28

ENT_HGW# show caller user jeremy@hgw.com

  User: jeremy@hgw.com, line Vi1, service PPP L2F, active 00:01:35
  PPP: LCP Open, CHAP (<- AAA), IPCP
  IP: Local 172.22.66.25, remote 172.30.2.1
  VPDN: NAS ISP_NAS, MID 1, MID open
        HGW  ENT_HGW, NAS CLID 36, HGW CLID 1, tunnel open
  Counts: 105 packets input, 8979 bytes, 0 no buffer
          0 input errors, 0 CRC, 0 frame, 0 overrun
          18 packets output, 295 bytes, 0 underruns
          0 output errors, 0 collisions, 0 interface resets
```

Step 4 From the tunnel server, ping Jeremy's PC at IP address 172.30.2.1:

```
ENT_HGW# ping 172.30.2.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.30.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 128/132/152 ms
```

Step 5 From the tunnel server, enter the **show interface virtual-access 1** command to verify that the interface is up, that LCP is open, and that no errors are reported:

```
ENT_HGW# show interface virtual-access 1
Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of FastEthernet0/0 (172.22.66.25)
  MTU 1500 bytes, BW 115 Kbit, DLY 100000 usec,
     reliablility 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Last input 00:00:02, output never, output hang never
  Last clearing of "show interface" counters 3d00h
  Queueing strategy: fifo
  Output queue 1/40, 0 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
     114 packets input, 9563 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
     27 packets output, 864 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

Step 6 From the tunnel server, display active tunnel statistics by entering the **show vpdn** command and the **show vpdn tunnel all** command:

```
ENT_HGW# show vpdn

% No active L2TP tunnels

L2F Tunnel and Session

 NAS CLID HGW CLID NAS Name        HGW Name        State
 36       1        ISP_NAS         ENT_HGW         open
                   172.22.66.23    172.22.66.25

 CLID   MID    Username                  Intf   State
 36     1      jeremy@hgw.com            Vi1    open

ENT_HGW# show vpdn tunnel all

% No active L2TP tunnels

L2F Tunnel
NAS name: ISP_NAS
NAS CLID: 36
NAS IP address 172.22.66.23
Gateway name: ENT_HGW
Gateway CLID: 1
Gateway IP address 172.22.66.25
State: open
Packets out: 52
Bytes out: 1799
Packets in: 100
Bytes in: 7143
```

Monitoring and Maintaining VPNs

To display useful information for monitoring and maintaining VPN sessions, use the following commands in privileged EXEC mode:

Command	Purpose
Router# **clear vpdn tunnel** [**pptp** | **l2f** | **l2tp**] *network-access-server gateway-name*	Shuts down a specific tunnel and all the sessions within the tunnel.
Router# **show interface virtual access** *number*	Displays information about the virtual access interface, LCP, protocol states, and interface statistics. The status of the virtual access interface should be: `Virtual-Access3 is up, line protocol is up`
Router# **show vpdn**	Displays a summary of all active VPN tunnels.
Router# **show vpdn domain**	Displays all VPN domains and DNIS groups configured on the NAS.

Command	Purpose
Router# **show vpdn group** [*name* | *name* **domain** | *name* **endpoint**]	Displays a summary of the relationships among VPDN groups and customer/VPDN profiles. When you include the name of the VPDN group, the output displays information on domain/DNIS, tunnel endpoint, session limits, group priority, active sessions, group status, and reserved sessions.
Router# **show vpdn history failure**	Displays information about VPN user failures.
Router# **show vpdn multilink**	Displays VPN multilink information.
Router# **show vpdn session** [**all** | **packets** | **sequence** | **state** | **timers** | **window**] [**interface** | **tunnel** | **username**]	Displays VPN session information including interface, tunnel, username, packets, status, and window statistics.
Router# **show vpdn tunnel** [**all** | **packets** | **state** | **summary** | **transport**] [**id** | **local-name** | **remote-name**]	Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.

Troubleshooting VPNs

Troubleshooting components in VPN is not always straightforward because there are multiple technologies and OSI layers involved. To display detailed messages about VPN and VPN-related events, use the following commands in EXEC mode:

Command	Purpose
Router# **debug aaa authentication**	Displays information on AAA authentication.
Router# **debug aaa authorization**	Displays information on AAA authorization.
Router# **debug ppp chap**	Displays CHAP packet exchanges.
Router# **debug ppp mppe**	Displays debug messages for MPPE events.
Router# **debug ppp negotiation**	Displays information about packets sent during PPP startup and detailed PPP negotiation options.
Router# **debug vpdn error**	Displays errors that prevent a tunnel from being established or errors that cause an established tunnel to be closed.
Router# **debug vpdn event**	Displays messages about events that are part of normal tunnel establishment or shutdown.
Router# **debug vpdn l2tp-sequencing**	Displays message about L2TP tunnel sequencing.
Router# **debug vpdn l2x-data**	Display messages about L2F and L2TP data information.
Router# **debug vpdn l2x-errors**	Displays L2F and L2TP protocol errors that prevent L2F and L2TP establishment or prevent normal operation.
Router# **debug vpdn l2x-events**	Displays messages about events that are part of normal tunnel establishment or shutdown for L2F and L2TP.
Router# **debug vpdn l2x-packets** or Router# **debug vpdn packet**	Displays each protocol packet exchanged. This option may result in a large number of debug messages and should generally be used only on a debug chassis with a single active session.

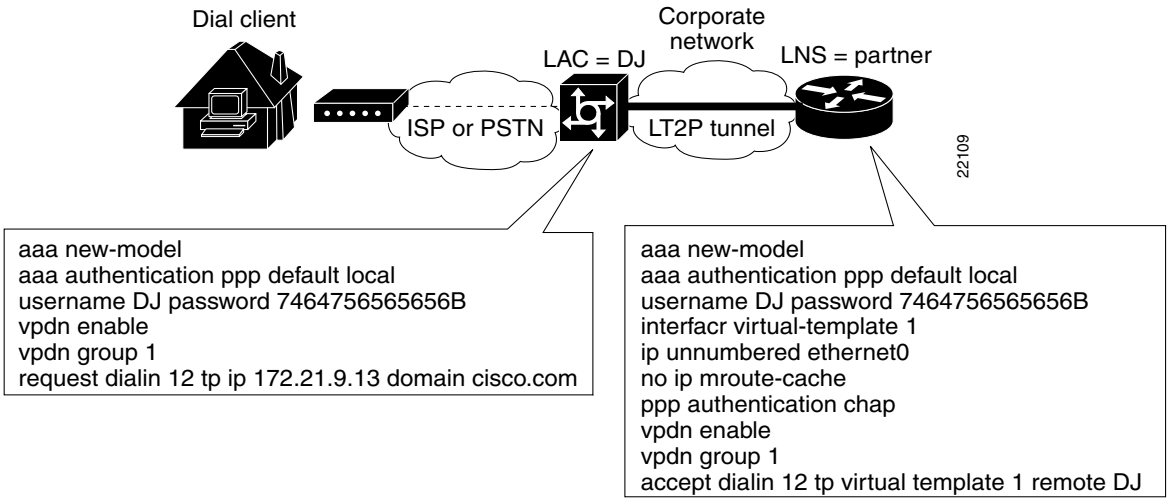
Successful Debug Examples

The following sections provide examples of debug output from successful VPN sessions:

- L2TP Dial-In Debug Output on NAS Example
- L2TP Dial-In Debug Output on a Tunnel Server Example
- L2TP Dial-Out Debug Output on a NAS Example
- L2TP Dial-Out Debug Output on a Tunnel Server Example

Figure 79 shows the topology used for the L2TP dial-in debug examples.

Figure 79 Topology Diagram for L2TP Dial-In Debug Example



L2TP Dial-In Debug Output on NAS Example

The following is debug output from a successful L2TP dial-in session on a NAS for the topology shown in Figure 79:

```
DJ# debug vpdn event

VPDN events debugging is on

DJ# debug vpdn l2x-events

L2X protocol events debugging is on

DJ# show debugging

VPN:
  L2X protocol events debugging is on
  VPDN events debugging is on
DJ#
20:47:33: %LINK-3-UPDOWN: Interface Async7, changed state to up
20:47:35: As7 VPDN: Looking for tunnel -- hoser.com --
20:47:35: As7 VPDN: Get tunnel info for hoser.com with NAS DJ, IP 172.21.9.13
20:47:35: As7 VPDN: Forward to address 172.21.9.13
20:47:35: As7 VPDN: Forwarding...
20:47:35: As7 VPDN: Bind interface direction=1
20:47:35: Tnl/Cl 8/1 L2TP: Session FS enabled
20:47:35: Tnl/Cl 8/1 L2TP: Session state change from idle to wait-for-tunnel
```

```
20:47:35: As7 8/1 L2TP: Create session
20:47:35: Tnl 8 L2TP: SM State idle
20:47:35: Tnl 8 L2TP: Tunnel state change from idle to wait-ctl-reply
20:47:35: Tnl 8 L2TP: SM State wait-ctl-reply
20:47:35: As7 VPDN: kath@hoser.com is forwarded
20:47:35: Tnl 8 L2TP: Got a challenge from remote peer, DJ
20:47:35: Tnl 8 L2TP: Got a response from remote peer, DJ
20:47:35: Tnl 8 L2TP: Tunnel Authentication success
20:47:35: Tnl 8 L2TP: Tunnel state change from wait-ctl-reply to established
20:47:35: Tnl 8 L2TP: SM State established
20:47:35: As7 8/1 L2TP: Session state change from wait-for-tunnel to wait-reply
20:47:35: As7 8/1 L2TP: Session state change from wait-reply to established
20:47:36: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async7, changed state to up
```

L2TP Dial-In Debug Output on a Tunnel Server Example

The following is debug output from a successful L2TP dial-in session on a tunnel server for the topology shown in Figure 79:

```
tunnel# debug vpdn l2x-events

L2X protocol events debugging is on

20:19:17: L2TP: I SCCRQ from DJ tnl 8
20:19:17: L2X: Never heard of DJ
20:19:17: Tnl 7 L2TP: New tunnel created for remote DJ, address 172.21.9.4
20:19:17: Tnl 7 L2TP: Got a challenge in SCCRQ, DJ
20:19:17: Tnl 7 L2TP: Tunnel state change from idle to wait-ctl-reply
20:19:17: Tnl 7 L2TP: Got a Challenge Response in SCCCN from DJ
20:19:17: Tnl 7 L2TP: Tunnel Authentication success
20:19:17: Tnl 7 L2TP: Tunnel state change from wait-ctl-reply to established
20:19:17: Tnl 7 L2TP: SM State established
20:19:17: Tnl/Cl 7/1 L2TP: Session FS enabled
20:19:17: Tnl/Cl 7/1 L2TP: Session state change from idle to wait-for-tunnel
20:19:17: Tnl/Cl 7/1 L2TP: New session created
20:19:17: Tnl/Cl 7/1 L2TP: O ICRP to DJ 8/1
20:19:17: Tnl/Cl 7/1 L2TP: Session state change from wait-for-tunnel to wait-connect
20:19:17: Tnl/Cl 7/1 L2TP: Session state change from wait-connect to established
20:19:17: Vi1 VPDN: Virtual interface created for kath@hoser.com
20:19:17: Vi1 VPDN: Set to Async interface
20:19:17: Vi1 VPDN: Clone from Vtemplate 1 filterPPP=0 blocking
20:19:18: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
20:19:18: Vi1 VPDN: Bind interface direction=2
20:19:18: Vi1 VPDN: PPP LCP accepting rcv CONFACK
20:19:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state
to up
```

L2TP Dial-Out Debug Output on a NAS Example

The following is sample output from the **debug dialer events** and **show debugging** EXEC commands for a successful dial-out session on a NAS:

```
NAS# debug dialer events

Dial on demand events debugging is on

NAS# show debugging

Dial on demand:
  Dial on demand events debugging is on
VPN:
```

```
  L2X protocol events debugging is on
  VPDN events debugging is on
NAS#
*Mar  1 00:05:26.155:%SYS-5-CONFIG_I:Configured from console by console
*Mar  1 00:05:26.899:%SYS-5-CONFIG_I:Configured from console by console
*Mar  1 00:05:36.195:L2TP:I SCCRQ from lns_l2x0 tnl 1
*Mar  1 00:05:36.199:Tnl 1 L2TP:New tunnel created for remote lns_l2x0, address
10.40.1.150
*Mar  1 00:05:36.203:Tnl 1 L2TP:Got a challenge in SCCRQ, lns_l2x0
*Mar  1 00:05:36.207:Tnl 1 L2TP:O SCCRP  to lns_l2x0 tnlid 1
*Mar  1 00:05:36.215:Tnl 1 L2TP:Tunnel state change from idle to wait-ctl-reply
*Mar  1 00:05:36.231:Tnl 1 L2TP:I SCCCN from lns_l2x0 tnl 1
*Mar  1 00:05:36.235:Tnl 1 L2TP:Got a Challenge Response in SCCCN from lns_l2x0
*Mar  1 00:05:36.239:Tnl 1 L2TP:Tunnel Authentication success
*Mar  1 00:05:36.239:Tnl 1 L2TP:Tunnel state change from wait-ctl-reply to established
*Mar  1 00:05:36.243:Tnl 1 L2TP:SM State established
*Mar  1 00:05:36.251:Tnl 1 L2TP:I OCRQ from lns_l2x0 tnl 1
*Mar  1 00:05:36.255:Tnl/Cl 1/1 L2TP:Session sequencing disabled
*Mar  1 00:05:36.259:Tnl/Cl 1/1 L2TP:Session FS enabled
*Mar  1 00:05:36.259:Tnl/Cl 1/1 L2TP:New session created
*Mar  1 00:05:36.263:12C:Same state, 0
*Mar  1 00:05:36.267:DSES 12C:Session create
*Mar  1 00:05:36.271:L2TP:Send OCRP
*Mar  1 00:05:36.275:Tnl/Cl 1/1 L2TP:Session state change from idle to wait-cs-answer
*Mar  1 00:05:36.279:DSES 0x12C:Building dialer map
*Mar  1 00:05:36.283:Dialout 0x12C:Next hop name is 71014
*Mar  1 00:05:36.287:Serial0:23 DDR:rotor dialout [priority]
*Mar  1 00:05:36.291:Serial0:23 DDR:Dialing cause dialer session 0x12C
*Mar  1 00:05:36.291:Serial0:23 DDR:Attempting to dial 71014
*Mar  1 00:05:36.479:%LINK-3-UPDOWN:Interface Serial0:22, changed state to up
*Mar  1 00:05:36.519:isdn_call_connect:Calling lineaction of Serial0:22
*Mar  1 00:05:36.519:Dialer0:Session free, 12C
*Mar  1 00:05:36.523::0 packets unqueued and discarded
*Mar  1 00:05:36.527:Se0:22 VPDN:Bind interface direction=1
*Mar  1 00:05:36.531:Se0:22 1/1 L2TP:Session state change from wait-cs-answer to
established
*Mar  1 00:05:36.531:L2TP:Send OCCN
*Mar  1 00:05:36.539:Se0:22 VPDN:bound to vpdn session
*Mar  1 00:05:36.555:Se0:22 1/1 L2TP:O FS failed
*Mar  1 00:05:36.555:Se0:22 1/1 L2TP:O FS failed
*Mar  1 00:05:42.515:%ISDN-6-CONNECT:Interface Serial0:22 is now connected to 71014
```

L2TP Dial-Out Debug Output on a Tunnel Server Example

The following is sample debug output from the **debug vpdn event**, **debug vpdn error**, **debug ppp chap**, **debug ppp negotiation**, and **debug dialer events** commands for a successful dial-out session on a tunnel server:

```
LNS# debug dialer events

Dial on demand events debugging is on

LNS# debug ppp negotiation

PPP protocol negotiation debugging is on

LNS# debug ppp chap

PPP authentication debugging is on

LNS# show debugging
```

```
Dial on demand:
  Dial on demand events debugging is on
PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
VPN:
  VPDN events debugging is on
  VPDN errors debugging is on
LNS#
*Apr 22 19:48:32.419:%SYS-5-CONFIG_I:Configured from console by console
*Apr 22 19:48:32.743:%SYS-5-CONFIG_I:Configured from console by console
*Apr 22 19:48:33.243:Di0 DDR:dialer_fsm_idle()
*Apr 22 19:48:33.271:Vi1 PPP:Phase is DOWN, Setup
*Apr 22 19:48:33.279:Vi1 PPP:Phase is DOWN, Setup
*Apr 22 19:48:33.279:Virtual-Access1 DDR:Dialing cause ip (s=10.60.1.160, d=10.10.1.110)
*Apr 22 19:48:33.279:Virtual-Access1 DDR:Attempting to dial 71014
*Apr 22 19:48:33.279:Tnl/Cl 1/1 L2TP:Session sequencing disabled
*Apr 22 19:48:33.279:Tnl/Cl 1/1 L2TP:Session FS enabled
*Apr 22 19:48:33.283:Tnl/Cl 1/1 L2TP:Session state change from idle to wait-for-tunnel
*Apr 22 19:48:33.283:Tnl/Cl 1/1 L2TP:Create dialout session
*Apr 22 19:48:33.283:Tnl 1 L2TP:SM State idle
*Apr 22 19:48:33.283:Tnl 1 L2TP:O SCCRQ
*Apr 22 19:48:33.283:Tnl 1 L2TP:Tunnel state change from idle to wait-ctl-reply
*Apr 22 19:48:33.283:Tnl 1 L2TP:SM State wait-ctl-reply
*Apr 22 19:48:33.283:Vi1 VPDN:Bind interface direction=2
*Apr 22 19:48:33.307:Tnl 1 L2TP:I SCCRP from lac_l2x0
*Apr 22 19:48:33.307:Tnl 1 L2TP:Got a challenge from remote peer, lac_l2x0
*Apr 22 19:48:33.307:Tnl 1 L2TP:Got a response from remote peer, lac_l2x0
*Apr 22 19:48:33.311:Tnl 1 L2TP:Tunnel Authentication success
*Apr 22 19:48:33.311:Tnl 1 L2TP:Tunnel state change from wait-ctl-reply to established
*Apr 22 19:48:33.311:Tnl 1 L2TP:O SCCCN  to lac_l2x0 tnlid 1
*Apr 22 19:48:33.311:Tnl 1 L2TP:SM State established
*Apr 22 19:48:33.311:L2TP:O OCRQ
*Apr 22 19:48:33.311:Vi1 1/1 L2TP:Session state change from wait-for-tunnel to wait-reply
*Apr 22 19:48:33.367:Vi1 1/1 L2TP:I OCRP from lac_l2x0 tnl 1, cl 0
*Apr 22 19:48:33.367:Vi1 1/1 L2TP:Session state change from wait-reply to wait-connect
*Apr 22 19:48:33.631:Vi1 1/1 L2TP:I OCCN from lac_l2x0 tnl 1, cl 1
*Apr 22 19:48:33.631:Vi1 1/1 L2TP:Session state change from wait-connect to established
*Apr 22 19:48:33.631:Vi1 VPDN:Connection is up, start LCP negotiation now
*Apr 22 19:48:33.631:%LINK-3-UPDOWN:Interface Virtual-Access1, changed state to up
*Apr 22 19:48:33.631:Vi1 DDR:dialer_statechange(), state=4Dialer statechange to up
Virtual-Access1
*Apr 22 19:48:33.631:Vi1 DDR:dialer_out_call_connected()
*Apr 22 19:48:33.631:Vi1 DDR:dialer_bind_profile() to Di0
*Apr 22 19:48:33.631:%DIALER-6-BIND:Interface Virtual-Access1 bound to profile
Dialer0Dialer call has been placed Virtual-Access1
*Apr 22 19:48:33.635:Vi1 PPP:Treating connection as a callout
*Apr 22 19:48:33.635:Vi1 PPP:Phase is ESTABLISHING, Active Open
*Apr 22 19:48:33.635:Vi1 LCP:O CONFREQ [Closed] id 1 len 15
*Apr 22 19:48:33.635:Vi1 LCP:   AuthProto CHAP (0x0305C22305)
*Apr 22 19:48:33.635:Vi1 LCP:   MagicNumber 0x50E7EC2A (0x050650E7EC2A)
*Apr 22 19:48:33.663:Vi1 LCP:I CONFREQ [REQsent] id 1 len 15
*Apr 22 19:48:33.663:Vi1 LCP:   AuthProto CHAP (0x0305C22305)
*Apr 22 19:48:33.663:Vi1 LCP:   MagicNumber 0x10820474 (0x050610820474)
*Apr 22 19:48:33.663:Vi1 LCP:O CONFACK [REQsent] id 1 len 15
*Apr 22 19:48:33.663:Vi1 LCP:   AuthProto CHAP (0x0305C22305)
*Apr 22 19:48:33.663:Vi1 LCP:   MagicNumber 0x10820474 (0x050610820474)
*Apr 22 19:48:33.663:Vi1 LCP:I CONFACK [ACKsent] id 1 len 15
*Apr 22 19:48:33.663:Vi1 LCP:   AuthProto CHAP (0x0305C22305)
*Apr 22 19:48:33.663:Vi1 LCP:   MagicNumber 0x50E7EC2A (0x050650E7EC2A)
*Apr 22 19:48:33.663:Vi1 LCP:State is Open
*Apr 22 19:48:33.663:Vi1 PPP:Phase is AUTHENTICATING, by both
*Apr 22 19:48:33.663:Vi1 CHAP:Using alternate hostname lns0
*Apr 22 19:48:33.663:Vi1 CHAP:O CHALLENGE id 1 len 25 from "lns0"
```

```
*Apr 22 19:48:33.679:Vi1 CHAP:I CHALLENGE id 1 len 35 from "user0@foo.com0"
*Apr 22 19:48:33.679:Vi1 AUTH:Started process 0 pid 92
*Apr 22 19:48:33.679:Vi1 CHAP:Using alternate hostname lns0
*Apr 22 19:48:33.683:Vi1 CHAP:O RESPONSE id 1 len 25 from "lns0"
*Apr 22 19:48:33.695:Vi1 CHAP:I SUCCESS id 1 len 4
*Apr 22 19:48:33.699:Vi1 CHAP:I RESPONSE id 1 len 35 from "user0@foo.com0"
*Apr 22 19:48:33.699:Vi1 CHAP:O SUCCESS id 1 len 4
*Apr 22 19:48:33.699:Vi1 DDR:dialer_remote_name() for user0@foo.com0
*Apr 22 19:48:33.699:Vi1 PPP:Phase is UP
*Apr 22 19:48:33.703:Vi1 IPCP:O CONFREQ [Closed] id 1 len 10
*Apr 22 19:48:33.703:Vi1 IPCP:   Address 10.20.1.150 (0x030614140196)
*Apr 22 19:48:33.703:Vi1 CCP:O CONFREQ [Closed] id 1 len 10
*Apr 22 19:48:33.703:Vi1 CCP:   LZSDCP history 1 check mode SEQ process UNCOMPRESSSED
(0x170600010201)
*Apr 22 19:48:33.711:Vi1 IPCP:I CONFREQ [REQsent] id 1 len 10
*Apr 22 19:48:33.715:Vi1 IPCP:   Address 10.20.1.120 (0x030614140178)
*Apr 22 19:48:33.715:Vi1 IPCP:O CONFACK [REQsent] id 1 len 10
*Apr 22 19:48:33.715:Vi1 IPCP:   Address 10.20.1.120 (0x030614140178)
*Apr 22 19:48:33.715:Vi1 CCP:I CONFREQ [REQsent] id 1 len 10
*Apr 22 19:48:33.715:Vi1 CCP:   LZSDCP history 1 check mode SEQ process UNCOMPRESSSED
(0x170600010201)
*Apr 22 19:48:33.715:Vi1 CCP:O CONFACK [REQsent] id 1 len 10
*Apr 22 19:48:33.715:Vi1 CCP:   LZSDCP history 1 check mode SEQ process UNCOMPRESSSED
(0x170600010201)
*Apr 22 19:48:33.719:Vi1 IPCP:I CONFACK [ACKsent] id 1 len 10
*Apr 22 19:48:33.719:Vi1 IPCP:   Address 10.20.1.150 (0x030614140196)
*Apr 22 19:48:33.719:Vi1 IPCP:State is Open
*Apr 22 19:48:33.719:Vi1 DDR:Dialer protocol up
*Apr 22 19:48:33.719:Dialer0:dialer_ckt_swt_client_connect:incoming circuit switched call
*Apr 22 19:48:33.719:Di0 IPCP:Install route to 10.20.1.120
*Apr 22 19:48:33.719:Vi1 CCP:I CONFACK [ACKsent] id 1 len 10
*Apr 22 19:48:33.719:Vi1 CCP:   LZSDCP history 1 check mode SEQ process UNCOMPRESSSED
(0x170600010201)
*Apr 22 19:48:33.719:Vi1 CCP:State is Open
*Apr 22 19:48:34.699:%LINEPROTO-5-UPDOWN:Line protocol on Interface Virtual-Access1,
changed state to up
```

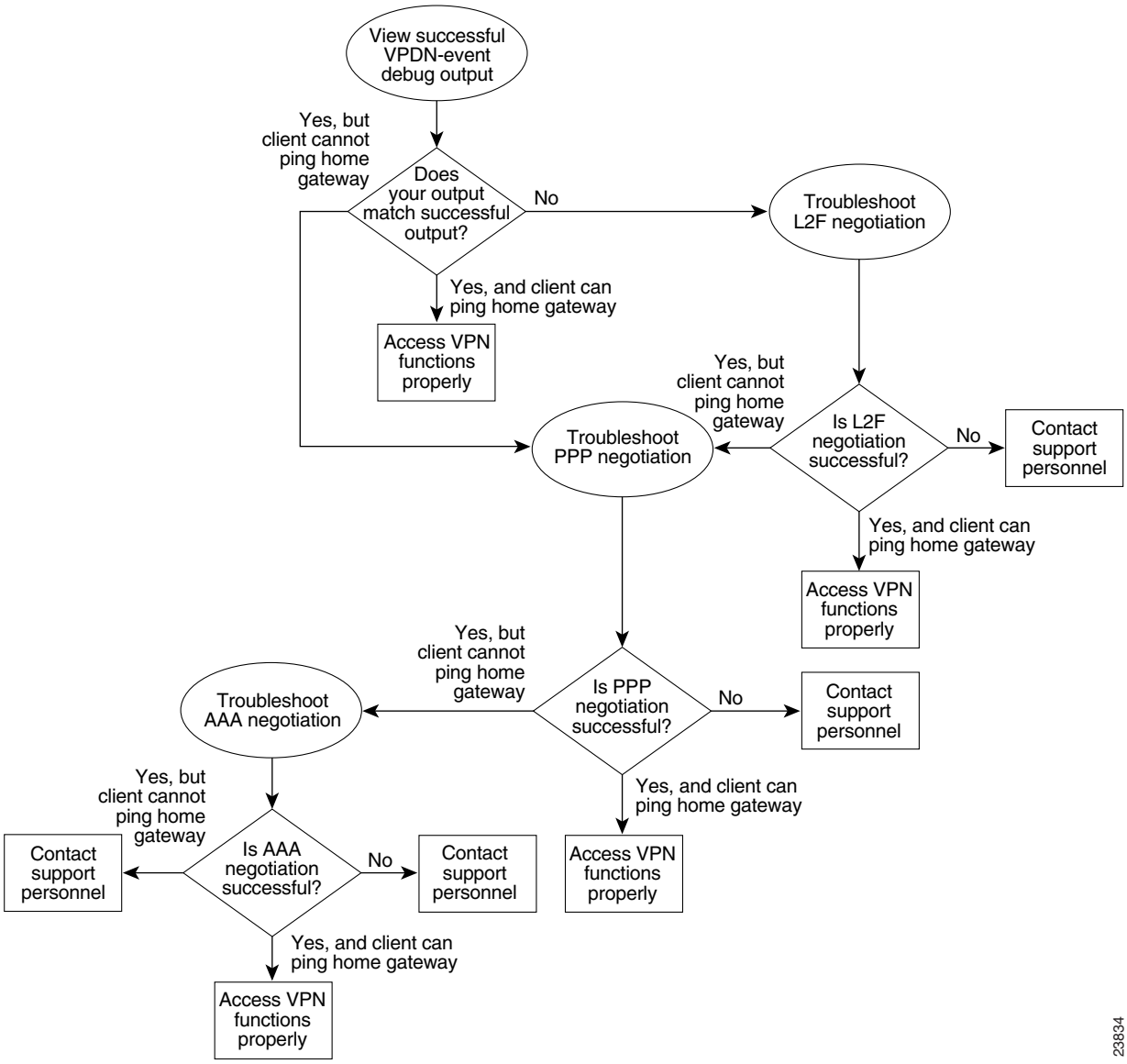
VPN Troubleshooting Methodology

This section describes a methodology for troubleshooting the VPN shown in Figure 80. First, view the debug output from a successful call. If your debug output does not match the successful output, follow the remaining steps to begin troubleshooting the network. The bolded lines of debug output indicate important information.

The following sections detail the steps involved in VPN troubleshooting:

- Comparing Your Debug Output to the Successful Debug Output
- Troubleshooting VPN Negotiation
- Troubleshooting PPP Negotiation
- Troubleshooting AAA Negotiation

Figure 80 Troubleshooting Flow Diagram for Access VPN with Remote AAA



If you are accessing the NAS and tunnel server through a Telnet connection, you need to enable the **terminal monitor** command. This command ensures that your EXEC session is receiving the logging and debug output from the devices.

When you finish troubleshooting, use the **undebug all** command to turn off all debug commands. Isolating debug output helps you efficiently build a network.

Comparing Your Debug Output to the Successful Debug Output

Enable the **debug vpdn-event** command on both the NAS and the tunnel server and dial in to the NAS. The following debug output shows successful VPN negotiation on the NAS and tunnel server:

```
NAS#
Jan  7 00:19:35.900: %LINK-3-UPDOWN: Interface Async9, changed state to up
Jan  7 00:19:39.532: sVPDN: Got DNIS string As9
Jan  7 00:19:39.532: As9 VPDN: Looking for tunnel -- hgw.com --
Jan  7 00:19:39.540: As9 VPDN: Get tunnel info for hgw.com with NAS ISP_NAS,
IP172.22.66.25
Jan  7 00:19:39.540: As9 VPDN: Forward to address 172.22.66.25
Jan  7 00:19:39.540: As9 VPDN: Forwarding...
Jan  7 00:19:39.540: As9 VPDN: Bind interface direction=1
Jan  7 00:19:39.540: As9 VPDN: jeremy@hgw.com is forwarded
Jan  7 00:19:40.540: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async9, changed state
to up

ENT_HGW#
Jan  7 00:19:39.967: VPDN: Chap authentication succeeded for ISP_NAS
Jan  7 00:19:39.967: Vi1 VPDN: Virtual interface created for jeremy@hgw.com
Jan  7 00:19:39.967: Vi1 VPDN: Set to Async interface
Jan  7 00:19:39.971: Vi1 VPDN: Clone from Vtemplate 1 filterPPP=0 blocking
6w5d: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
Jan  7 00:19:40.051: Vi1 VPDN: Bind interface direction=2
Jan  7 00:19:40.051: Vi1 VPDN: PPP LCP accepted rcv CONFACK
Jan  7 00:19:40.051: Vi1 VPDN: PPP LCP accepted sent CONFACK
6w5d: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
```

If you see the debug output shown but cannot ping the tunnel server, go to the next section, "Troubleshooting PPP Negotiation."

If you do not see the above debug output, go to the section "Troubleshooting VPN Negotiation" later in this chapter.

Troubleshooting VPN Negotiation

The following sections describe several common misconfigurations that prevent successful VPN (either L2F or L2TP) negotiation:

- Misconfigured NAS Tunnel Secret
- Misconfigured Tunnel Server Tunnel Secret
- Misconfigured Tunnel Name
- Control Packet Problem on the NAS

Misconfigured NAS Tunnel Secret

The NAS and the tunnel server must both have the same usernames with the same password to authenticate the L2F tunnel. These usernames are called the tunnel secret. In this scenario, these usernames are ISP_NAS and ENT_HGW. The password is cisco for both usernames on both systems.

If one of the tunnel secrets on the NAS is incorrect, you will see the following debug output when you dial in to the NAS and the **debug vpdn l2x-errors** command is enabled on the NAS and tunnel server:

```
NAS#
Jan  1 00:26:49.899: %LINK-3-UPDOWN: Interface Async3, changed state to up
Jan  1 00:26:54.643: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async3, cha
nged state to up
Jan  1 00:27:00.559: L2F: Resending L2F_OPEN, time #1
```

```
Jan  1 00:27:05.559: L2F: Resending L2F_ECHO, time #1
Jan  1 00:27:05.559: L2F: Resending L2F_OPEN, time #2
Jan  1 00:27:10.559: L2F: Resending L2F_ECHO, time #2
Jan  1 00:27:10.559: L2F: Resending L2F_OPEN, time #3
Jan  1 00:27:15.559: L2F: Resending L2F_ECHO, time #3
Jan  1 00:27:15.559: L2F: Resending L2F_OPEN, time #4
Jan  1 00:27:20.559: L2F: Resending L2F_ECHO, time #4
Jan  1 00:27:20.559: L2F: Resending L2F_OPEN, time #5
Jan  1 00:27:25.559: L2F: Resending L2F_ECHO, time #5
Jan  1 00:27:25.559: L2F: Resend packet (type 2) around too long, time to kill off the
tunnel
NAS#

ENT_HGW#
Jan  1 00:26:53.645: L2F: Packet has bogus2 key C8353FAB B6369121
5w6d: %VPDN-6-AUTHENFAIL: L2F HGW , authentication failure for  tunnel ISP_NAS; Invalid
key
5w6d: %VPDN-5-UNREACH: L2F NAS 172.22.66.23 is unreachable
Jan  1 00:27:00.557: L2F: Gateway received tunnel OPEN while in state closed
ENT_HGW#
```

The phrase “time to kill off the tunnel” in the NAS debug output indicates that the tunnel was not opened. The phrase “Packet has bogus2 key” in the tunnel server debug output indicates that the NAS has an incorrect tunnel secret.

To avoid this problem, make sure that you configure both the NAS and tunnel server for the same two tunnel secret usernames with the same password.

Misconfigured Tunnel Server Tunnel Secret

If one of the tunnel secret usernames on the tunnel server is incorrect, the following debug output appears when you dial in to the NAS and the **debug vpdn l2x-errors** command is enabled on the NAS and tunnel server:

```
NAS#
Jan  1 00:45:27.123: %LINK-3-UPDOWN: Interface Async7, changed state to up
Jan  1 00:45:30.939: L2F: Packet has bogus1 key B6C656EE 5FAC6B3
Jan  1 00:45:30.939: %VPDN-6-AUTHENFAIL: L2F NAS ISP_NAS, authentication failure
 for  tunnel ENT_HGW; Invalid key
Jan  1 00:45:31.935: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async7, cha
nged state to up
Jan  1 00:45:35.559: L2F: Resending L2F_OPEN, time #1
Jan  1 00:45:35.559: L2F: Packet has bogus1 key B6C656EE 5FAC6B3

ENT_HGW#
Jan  1 00:45:30.939: L2F: Tunnel authentication succeeded for ISP_NAS
Jan  1 00:45:35.559: L2F: Gateway received tunnel OPEN while in state open
Jan  1 00:45:40.559: L2F: Gateway received tunnel OPEN while in state open
Jan  1 00:45:45.559: L2F: Gateway received tunnel OPEN while in state open
Jan  1 00:45:50.559: L2F: Gateway received tunnel OPEN while in state open
```

Notice how this output is similar to the debug output you see when the NAS has a misconfigured tunnel secret username. This time you see the phrase “Packet has bogus1 key” on the NAS instead of the tunnel server. This phrase tells you that the tunnel server has an incorrect tunnel secret username.

To avoid this problem, make sure that you configure both the NAS and tunnel server for the same two tunnel secret usernames with the same password.

Misconfigured Tunnel Name

If the NAS and tunnel server do not have matching tunnel names, they cannot establish an L2F tunnel. On the tunnel server, these tunnel names are configured under the **vpdn-group 1** command by using the **local name** command. On the NAS, these names are configured on the RADIUS server.

The tunnel server must be configured to accept tunnels from the name that the NAS sends it. This is done using the **accept-dialin l2f virtual-template 1 remote ISP_NAS** command, where **ISP_NAS** is the name. The name it returns to the NAS is configured using the **local name ENT_HGW** command, where **ENT_HGW** is the name. These commands appear in the following running configuration:

```
vpdn-group 1
 accept-dialin l2f virtual-template 1 remote ISP_NAS
 local name ENT_HGW
```

On the RADIUS server, the tunnel names are configured by adding profiles to the **NAS_Group** group with the names **ISP_NAS** and **ENT_HGW**.

In the following debug output, the NAS attempted to open a tunnel using the name isp. Because the tunnel server did not know this name, it did not open the tunnel. To see the following debug output, enable the **debug vpdn l2x-events** and **debug vpdn l2x-errors** commands on the tunnel server:

```
ENT_HGW#
Jan  1 01:28:54.207: L2F: L2F_CONF received
Jan  1 01:28:54.207: L2X: Never heard of isp
Jan  1 01:28:54.207: L2F: Couldn't find tunnel named isp
```

To avoid the problem described, make sure that the tunnel names match on the tunnel server and on the RADIUS server.

Control Packet Problem on the NAS

The following example assumes that you suspect an error in parsing control packets. You can use the **debug vpdn packet** command with the **control** keyword to verify control packet information.

```
ISP_NAS# debug vpdn packet control

20:50:27: %LINK-3-UPDOWN: Interface Async7, changed state to up
20:50:29: Tnl 9 L2TP: O SCCRQ
20:50:29: Tnl 9 L2TP: O SCCRQ, flg TLF, ver 2, len 131, tnl 0, cl 0, ns 0, nr 0
20:50:29: contiguous buffer, size 131
         C8 02 00 83 00 00 00 00 00 00 00 80 08 00 00
         00 00 00 01 80 08 00 00 00 02 01 00 80 0A 00 00
         00 03 00 00 00 03 80 0A 00 00 00 04 00 00 00 ...
20:50:29: Tnl 9 L2TP: Parse AVP 0, len 8, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Parse SCCRP
20:50:29: Tnl 9 L2TP: Parse AVP 2, len 8, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Protocol Ver 256
20:50:29: Tnl 9 L2TP: Parse AVP 3, len 10, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Framing Cap 0x0x3
20:50:29: Tnl 9 L2TP: Parse AVP 4, len 10, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Bearer Cap 0x0x3
20:50:29: Tnl 9 L2TP: Parse AVP 6, len 8, flag 0x0x0
20:50:29: Tnl 9 L2TP: Firmware Ver 0x0x1120
20:50:29: Tnl 9 L2TP: Parse AVP 7, len 12, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Hostname DJ
20:50:29: Tnl 9 L2TP: Parse AVP 8, len 25, flag 0x0x0
20:50:29: Tnl 9 L2TP: Vendor Name Cisco Systems, Inc.
20:50:29: Tnl 9 L2TP: Parse AVP 9, len 8, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Assigned Tunnel ID 8
20:50:29: Tnl 9 L2TP: Parse AVP 10, len 8, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Rx Window Size 4
```

```
20:50:29: Tnl 9 L2TP: Parse AVP 11, len 22, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Chlng D807308D106259C5933C6162ED3A1689
20:50:29: Tnl 9 L2TP: Parse AVP 13, len 22, flag 0x0x8000 (M)
20:50:29: Tnl 9 L2TP: Chlng Resp 9F6A3C70512BD3E2D44DF183C3FFF2D1
20:50:29: Tnl 9 L2TP: No missing AVPs in SCCRP
20:50:29: Tnl 9 L2TP: Clean Queue packet 0
20:50:29: Tnl 9 L2TP: I SCCRP, flg TLF, ver 2, len 153, tnl 9, cl 0, ns 0, nr 1
 contiguous pak, size 153
         C8 02 00 99 00 09 00 00 00 00 00 01 80 08 00 00
         00 00 00 02 80 08 00 00 00 02 01 00 80 0A 00 00
         00 03 00 00 00 03 80 0A 00 00 00 04 00 00 00 ...
20:50:29: Tnl 9 L2TP: I SCCRP from DJ
20:50:29: Tnl 9 L2TP: O SCCCN  to DJ tnlid 8
20:50:29: Tnl 9 L2TP: O SCCCN, flg TLF, ver 2, len 42, tnl 8, cl 0, ns 1, nr 1
20:50:29: contiguous buffer, size 42
         C8 02 00 2A 00 08 00 00 00 01 00 01 80 08 00 00
         00 00 00 03 80 16 00 00 00 0D 4B 2F A2 50 30 13
         E3 46 58 D5 35 8B 56 7A E9 85
20:50:29: As7 9/1 L2TP: O ICRQ to DJ 8/0
20:50:29: As7 9/1 L2TP: O ICRQ, flg TLF, ver 2, len 48, tnl 8, cl 0, ns 2, nr 1
20:50:29: contiguous buffer, size 48
         C8 02 00 30 00 08 00 00 00 02 00 01 80 08 00 00
         00 00 00 0A 80 08 00 00 00 0E 00 01 80 0A 00 00
         00 0F 00 00 00 04 80 0A 00 00 00 12 00 00 00 ...
20:50:29: Tnl 9 L2TP: Clean Queue packet 1
20:50:29: Tnl 9 L2TP: Clean Queue packet 2
20:50:29: Tnl 9 L2TP: I ZLB ctrl ack, flg TLF, ver 2, len 12, tnl 9, cl 0, ns 1, nr 2
 contiguous pak, size 12
         C8 02 00 0C 00 09 00 00 00 01 00 02
20:50:30: As7 9/1 L2TP: Parse AVP 0, len 8, flag 0x0x8000 (M)
20:50:30: As7 9/1 L2TP: Parse ICRP
20:50:30: As7 9/1 L2TP: Parse AVP 14, len 8, flag 0x0x8000 (M)
20:50:30: As7 9/1 L2TP: Assigned Call ID 1
20:50:30: As7 9/1 L2TP: No missing AVPs in ICRP
20:50:30: Tnl 9 L2TP: Clean Queue packet 2
20:50:30: As7 9/1 L2TP: I ICRP, flg TLF, ver 2, len 28, tnl 9, cl 1, ns 1, nr 3
 contiguous pak, size 28
         C8 02 00 1C 00 09 00 01 00 01 00 03 80 08 00 00
         00 00 00 0B 80 08 00 00 00 0E 00 01
20:50:30: As7 9/1 L2TP: O ICCN to DJ 8/1
20:50:30: As7 9/1 L2TP: O ICCN, flg TLF, ver 2, len 203, tnl 8, cl 1, ns 3, nr 2
20:50:30: contiguous buffer, size 203
         C8 02 00 CB 00 08 00 01 00 03 00 02 80 08 00 00
         00 00 00 0C 80 0A 00 00 00 18 00 00 DA C0 80 0A
         00 00 00 13 00 00 00 02 00 28 00 00 00 1B 02 ...
20:50:30: Tnl 9 L2TP: Clean Queue packet 3
20:50:30: As7 9/1 L2TP: I ZLB ctrl ack, flg TLF, ver 2, len 12, tnl 9, cl 1, ns 2, nr 4
 contiguous pak, size 12
         C8 02 00 0C 00 09 00 01 00 02 00 04
20:50:30: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async7, changed state to up
```

If you fixed the problem in your configuration, return to the section "Verifying VPN Sessions" earlier in this chapter.

If your call still cannot successfully complete L2F negotiation, contact your support personnel.

Troubleshooting PPP Negotiation

This section first shows debug output of successful PPP negotiation. The subsequent sections explain several common problems that prevent successful PPP negotiation:

- Successful PPP Negotiation Debug Output
- Non-Cisco Device Connectivity Problem
- Mismatched Username Example

Enable the **debug ppp negotiation** command on the tunnel server and dial in to the NAS.

Successful PPP Negotiation Debug Output

The following debug output shows successful PPP negotiation on the tunnel server:

```
1d02h: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Feb  4 14:14:40.505: Vi1 PPP: Treating connection as a dedicated line
*Feb  4 14:14:40.505: Vi1 PPP: Phase is ESTABLISHING, Active Open
*Feb  4 14:14:40.505: Vi1 PPP: Treating connection as a dedicated line
*Feb  4 14:14:40.505: Vi1 PPP: Phase is AUTHENTICATING, by this end
*Feb  4 14:14:40.509: Vi1 PPP: Phase is UP
```

If your call successfully completed PPP negotiation, but you still cannot ping the tunnel server, go to the section "Troubleshooting AAA Negotiation" later in this chapter.

Non-Cisco Device Connectivity Problem

The **debug ppp authentication** and **debug ppp negotiation** commands are enabled to decipher a CHAP negotiation problem. This is due to a connectivity problem between a Cisco and non-Cisco device. Also note that the **service-timestamps** command is enabled on the router. The **service-timestamps** command is helpful to decipher timing and keepalive issues, and we recommend that you always enable this command.

```
Router# debug ppp authentication

PPP authentication debugging is on

Router# debug ppp negotiation

PPP protocol negotiation debugging is on
3:22:53: ppp: sending CONFREQ, type = 3 (CI_AUTHTYPE), value = C223/5
3:22:53: ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = C6091F.
3:22:55: ppp: sending CONFREQ, type = 3 (CI_AUTHTYPE), value = C223/5
3:22:55: ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = C6091F
3:22:55: PPP BRI0: B-Channel 1: received config for type = 0x0 (??)
3:22:55: PPP BRI0: B-Channel 1: rcvd unknown option 0x0 rejected
3:22:55: PPP BRI0: B-Channel 1: received config for type = 0x1 (MRU) value = 0x5
F4 rejected
3:22:55: PPP BRI0: B-Channel 1: received config for type = 0x3 (AUTHTYPE) value
= 0xC223 value = 0x5 acked
3:22:55: PPP BRI0: B-Channel 1: received config for type = 0x11 (MULTILINK_MRRU)
rejected
3:22:55: PPP BRI0: B-Channel 1: received config for type = 0x13 (UNKNOWN)
3:22:55: PPP BRI0: B-Channel 1: rcvd unknown option 0x13 rejected
3:22:55: ppp: config REJ received, type = 3 (CI_AUTHTYPE), value = C223/5
3:22:55: ppp: sending CONFREQ, type = 3 (CI_AUTHTYPE), value = C223/5
3:22:55: ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = C6091F
3:22:55: PPP BRI0: B-Channel 1: received config for type = 0x3 (AUTHTYPE) value= 0xC2.
Success rate is 0 percent (0/5)
moog#23 value = 0x5 acked
```

```
3:22:55: ppp: config REJ received, type = 3 (CI_AUTHTYPE), value = C223/5

3:22:55: ppp: BRI0: B-Channel 1 closing connection because remote won't authenticate

3:22:55: ppp: sending CONFREQ, type = 3 (CI_AUTHTYPE), value = C223/5
3:22:55: ppp: sending CONFREQ, type = 5 (CI_MAGICNUMBER), value = C6091F
3:22:55: %ISDN-6-DISCONNECT: Interface BRI0: B-Channel 1  disconnected from 0123
5820040 , call lasted 2 seconds
3:22:56: %LINK-3-UPDOWN: Interface BRI0: B-Channel 1, changed state to down
Indication:
```

Mismatched Username Example

The following **debug ppp chap** sample output excerpt shows a CHAP authentication failure caused by a configuration mismatch between devices. Verifying and correcting any username and password mismatch should remedy this problem.

```
Router# debug ppp chap

ppp: received conf.ig for type = 5 (MAGICNUMBER) value = 1E24718 acked
PPP BRI0: B-Channel 1: state = ACKSENT fsm_rconfack(C021): rcvd id E6
ppp: config ACK received, type = 3 (CI_AUTHTYPE), value = C223
ppp: config ACK received, type = 5 (CI_MAGICNUMBER), value = 28CEF76C
BRI0: B-Channel 1: PPP AUTH CHAP input code = 1 id = 83 len = 16
BRI0: B-Channel 1: PPP AUTH CHAP input code = 2 id = 96 len = 28
BRI0: B-Channel 1: PPP AUTH CHAP input code = 4 id = 83 len = 21
BRI0: B-Channel 1: Failed CHAP authentication with remote.
Remote message is: MD compare failed
```

If your call cannot successfully complete PPP negotiation, contact your support personnel.

Troubleshooting AAA Negotiation

This section first shows debug output of successful AAA negotiation. The subsequent sections explain several common misconfigurations that prevent successful AAA negotiation:

- Successful AAA Negotiation
- Incorrect User Password
- Error Contacting RADIUS Server
- Misconfigured AAA Authentication

Successful AAA Negotiation

Enable the **debug aaa authentication** and **debug aaa authorization** commands on the tunnel server and dial in to the NAS.

The following debug output shows successful AAA negotiation on the tunnel server. This output has been edited to exclude repetitive lines.

```
ENT_HGW#
Jan  7 19:29:44.132: AAA/AUTHEN: create_user (0x612D550C) user='ENT_HGW' ruser='
' port='' rem_addr='' authen_type=CHAP service=PPP priv=1
Jan  7 19:29:44.132: AAA/AUTHEN/START (384300079): port='' list='default' action
=SENDAUTH service=PPP
Jan  7 19:29:44.132: AAA/AUTHEN/START (384300079): found list default
Jan  7 19:29:44.132: AAA/AUTHEN/START (384300079): Method=LOCAL
Jan  7 19:29:44.132: AAA/AUTHEN (384300079): status = PASS
Jan  7 19:29:44.132: AAA/AUTHEN: create_user (0x612D550C) user='ISP_NAS' ruser='
' port='' rem_addr='' authen_type=CHAP service=PPP priv=1
```

```
Jan  7 19:29:44.132: AAA/AUTHEN/START (2545876944): port='' list='default' actio
n=SENDAUTH service=PPP
Jan  7 19:29:44.132: AAA/AUTHEN/START (2545876944): found list default
Jan  7 19:29:44.132: AAA/AUTHEN/START (2545876944): Method=LOCAL
Jan  7 19:29:44.132: AAA/AUTHEN (2545876944): status = PASS
Jan  7 19:29:44.228: AAA/AUTHEN: create_user (0x612F1F78) user='jeremy@hgw.com'
ruser='' port='Virtual-Access1' rem_addr='408/5550945' authen_type=CHAP service=
PPP priv=1
Jan  7 19:29:44.228: AAA/AUTHEN/START (101773535): port='Virtual-Access1' list=''
action=LOGIN service=PPP
Jan  7 19:29:44.228: AAA/AUTHEN/START (101773535): using "default" list
Jan  7 19:29:44.228: AAA/AUTHEN/START (101773535): Method=LOCAL
Jan  7 19:29:44.228: AAA/AUTHEN (101773535): status = ERROR
Jan  7 19:29:44.228: AAA/AUTHEN/START (101773535): Method=RADIUS
Jan  7 19:29:44.692: AAA/AUTHEN (101773535): status = PASS
Jan  7 19:29:44.692: Vi1 AAA/AUTHOR/LCP: Authorize LCP
Jan  7 19:29:44.692: AAA/AUTHOR/LCP Vi1 (3630870259): Port='Virtual-Access1' list=''
service=NET
Jan  7 19:29:44.692: AAA/AUTHOR/LCP: Vi1 (3630870259) user='jeremy@hgw.com'
Jan  7 19:29:44.692: AAA/AUTHOR/LCP: Vi1 (3630870259) send AV service=ppp
Jan  7 19:29:44.692: AAA/AUTHOR/LCP: Vi1 (3630870259) send AV protocol=lcp
Jan  7 19:29:44.692: AAA/AUTHOR/LCP (3630870259) found list "default"
Jan  7 19:29:44.692: AAA/AUTHOR/LCP: Vi1 (3630870259) Method=RADIUS
Jan  7 19:29:44.692: AAA/AUTHOR (3630870259): Post authorization status = PASS_REPL
Jan  7 19:29:44.696: Vi1 AAA/AUTHOR/FSM: We can start IPCP
6w5d: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
Jan  7 19:29:47.792: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 172.30.2.1
```

If the above debug output appears, but you still cannot ping the tunnel server, contact your support personnel and troubleshoot your network backbone.

If you did not see the debug output above, you need to troubleshoot AAA negotiation.

Incorrect User Password

If the user password is incorrect (or it is incorrectly configured), the tunnel will be established, but the tunnel server will not authenticate the user. If the user password is incorrect, the following debug output appears on the NAS and tunnel server when you dial in to the NAS and the **debug vpdn l2x-errors** and **debug vpdn l2x-events** commands are enabled:

```
ISP_NAS#
Jan  1 01:00:01.555: %LINK-3-UPDOWN: Interface Async12, changed state to up
Jan  1 01:00:05.299:  L2F: Tunnel state closed
Jan  1 01:00:05.299:  L2F: MID  state closed
Jan  1 01:00:05.299: L2F: Open UDP socket to 172.22.66.25
Jan  1 01:00:05.299:  L2F: Tunnel state opening
Jan  1 01:00:05.299: As12 L2F: MID jeremy@hgw.com state waiting_for_tunnel
Jan  1 01:00:05.303: L2F: L2F_CONF received
Jan  1 01:00:05.303: L2F: Removing resend packet (L2F_CONF)
Jan  1 01:00:05.303: ENT_HGW L2F: Tunnel state open
Jan  1 01:00:05.307: L2F: L2F_OPEN received
Jan  1 01:00:05.307: L2F: Removing resend packet (L2F_OPEN)
Jan  1 01:00:05.307: L2F: Building nas2gw_mid0
Jan  1 01:00:05.307: L2F: L2F_CLIENT_INFO: CLID/DNIS 4089548021/5550945
Jan  1 01:00:05.307: L2F: L2F_CLIENT_INFO: NAS-Port Async12
Jan  1 01:00:05.307: L2F: L2F_CLIENT_INFO: Client-Bandwidth-Kbps 115
Jan  1 01:00:05.307: L2F: L2F_CLIENT_INFO: NAS-Rate L2F/26400/28800
Jan  1 01:00:05.307: As12 L2F: MID jeremy@hgw.com state opening
Jan  1 01:00:05.307: L2F: Tunnel authentication succeeded for ENT_HGW
Jan  1 01:00:05.391: L2F: L2F_OPEN received
Jan  1 01:00:05.391: L2F: Got a MID management packet
Jan  1 01:00:05.391: L2F: Removing resend packet (L2F_OPEN)
Jan  1 01:00:05.391: As12 L2F: MID jeremy@hgw.com state open
```

```
Jan  1 01:00:05.391: As12 L2F: MID synced NAS/HG Clid=47/12 Mid=1
Jan  1 01:00:05.523: L2F: L2F_CLOSE received
Jan  1 01:00:05.523: %VPDN-6-AUTHENERR: L2F HGW ENT_HGW cannot locate a AAA server for
As12 user jeremy@hgw.com; Authentication failure

ENT_HGW#
Jan  1 01:00:05.302: L2F: L2F_CONF received
Jan  1 01:00:05.302: L2F: Creating new tunnel for ISP_NAS
Jan  1 01:00:05.302:  L2F: Tunnel state closed
Jan  1 01:00:05.302: L2F: Got a tunnel named ISP_NAS, responding
Jan  1 01:00:05.302: L2F: Open UDP socket to 172.22.66.23
Jan  1 01:00:05.302: ISP_NAS L2F: Tunnel state opening
Jan  1 01:00:05.306: L2F: L2F_OPEN received
Jan  1 01:00:05.306: L2F: Removing resend packet (L2F_CONF)
Jan  1 01:00:05.306: ISP_NAS L2F: Tunnel state open
Jan  1 01:00:05.306: L2F: Tunnel authentication succeeded for ISP_NAS
Jan  1 01:00:05.310: L2F: L2F_OPEN received
Jan  1 01:00:05.310: L2F: L2F_CLIENT_INFO: CLID/DNIS 4089548021/5550945
Jan  1 01:00:05.310: L2F: L2F_CLIENT_INFO: NAS-Port Async12
Jan  1 01:00:05.310: L2F: L2F_CLIENT_INFO: Client-Bandwidth-Kbps 115
Jan  1 01:00:05.310: L2F: L2F_CLIENT_INFO: NAS-Rate L2F/26400/28800
Jan  1 01:00:05.310: L2F: Got a MID management packet
Jan  1 01:00:05.310:  L2F: MID  state closed
Jan  1 01:00:05.310: L2F: Start create mid intf process for jeremy@hgw.com
5w6d: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
Jan  1 01:00:05.390: Vi1 L2X: Discarding packet because of no mid/session
Jan  1 01:00:05.390: Vi1 L2F: Transfer NAS-Rate L2F/26400/28800 to LCP
Jan  1 01:00:05.390: Vi1 L2F: Finish create mid intf for jeremy@hgw.com
Jan  1 01:00:05.390: Vi1 L2F: MID jeremy@hgw.com state open
5w6d: %VPDN-6-AUTHENERR: L2F HGW ENT_HGW cannot locate a AAA server for Vi1 user
jeremy@hgw.com; Authentication failure
```

Error Contacting RADIUS Server

If the **aaa authorization** command on the tunnel server is configured with the **default radius none** keywords, the tunnel server may allow unauthorized access to your network.

This command is an instruction to first use RADIUS for authorization. The tunnel server first contacts the RADIUS server (because of the **radius** keyword). If an error occurs when the tunnel server contacts the RADIUS server, the tunnel server does not authorize the user (because of the **none** keyword).

To see the following debug output, enable the **debug aaa authorization** command on the tunnel server and dial in to the NAS:

```
ENT_HGW#
*Feb  5 17:27:36.166: Vi1 AAA/AUTHOR/LCP: Authorize LCP
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP Vi1 (3192359105): Port='Virtual-Access1' list=''
service=NET
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP: Vi1 (3192359105) user='jeremy@hgw.com'
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP: Vi1 (3192359105) send AV service=ppp
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP: Vi1 (3192359105) send AV protocol=lcp
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP (3192359105) found list "default"
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP: Vi1 (3192359105) Method=RADIUS
*Feb  5 17:27:36.166: AAA/AUTHOR (3192359105): Post authorization status = ERROR
*Feb  5 17:27:36.166: AAA/AUTHOR/LCP: Vi1 (3192359105) Method=NONE
*Feb  5 17:27:36.166: AAA/AUTHOR (3192359105): Post authorization status = PASS_ADD
*Feb  5 17:27:36.166: Vi1 CHAP: O SUCCESS id 1 len 4
```

Caution Using the **none** keyword can allow unauthorized access to your network. Because of the risk of such errors occurring, we strongly recommend that you do not use the **none** keyword in your **aaa** commands.

Misconfigured AAA Authentication

If you reverse the order of the **local** and **radius** keywords in the **aaa authentication ppp** command on the tunnel server, the L2F tunnel cannot be established. The command should be configured as **aaa authentication ppp default local radius**.

If you configure the command as **aaa authentication ppp default radius local**, the tunnel server first tries to authenticate the L2F tunnel using RADIUS. The RADIUS server sends the following message to the tunnel server. To see this message, enable the **debug radius** command.

```
ENT_HGW#
Jan  1 01:34:47.827: RADIUS: SENDPASS not supported (action=4)
```

The RADIUS protocol does not support inbound challenges. This means that RADIUS is designed to authenticate user information, but it is not designed to be authenticated by others. When the tunnel server requests the tunnel secret from the RADIUS server, it responds with the "SENDPASS not supported" message.

To avoid this problem, use the **aaa authentication ppp default local radius** command on the tunnel server.

If your call still cannot successfully complete AAA negotiation, contact your support personnel.

Configuration Examples for VPN

This section provides the following configuration examples:

- Client-Initiated Dial-In Configuration Example
- VPN Tunnel Authentication Examples
- NAS Comprehensive Dial-In Configuration Example
- Tunnel Server Comprehensive Dial-in Configuration Example
- NAS Configured for Both Dial-In and Dial-Out Example
- Tunnel Server Configured for Both Dial-In and Dial-Out Example
- RADIUS Profile Examples
- TACACS+ Profile Examples

Client-Initiated Dial-In Configuration Example

The following example shows the running configuration of a tunnel server configured for PPTP using an ISA card to perform 40-bit MPPE encryption. It does not have an AAA configuration.

```
Current configuration
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname PNS
!
no logging console guaranteed
enable password lab
!
```

```
username tester41 password 0 lab41
!
ip subnet-zero
no ip domain-lookup
!
vpdn enable
!
vpdn-group 1
! Default PPTP VPDN group
 accept-dialin
  protocol pptp
  virtual-template 1
 local name cisco_pns
!
memory check-interval 1
!
controller ISA 5/0
 encryption mppe
!
process-max-time 200
!
interface FastEthernet0/0
 ip address 10.1.1.12 255.255.255.0
 no ip directed-broadcast
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 10.1.2.12 255.255.255.0
 no ip directed-broadcast
 duplex auto
 speed auto
!
interface Serial1/0
 no ip address
 no ip directed-broadcast
 shutdown
 framing c-bit
 cablelength 10
 dsu bandwidth 44210
!
interface Serial1/1
 no ip address
 no ip directed-broadcast
 shutdown
 framing c-bit
 cablelength 10
 dsu bandwidth 44210
!
interface FastEthernet4/0
 no ip address
 no ip directed-broadcast
 shutdown
 duplex half
!
interface Virtual-Template1
 ip unnumbered FastEthernet0/0
 no ip directed-broadcast
 ip mroute-cache
 no keepalive
 ppp encrypt mppe 40
 ppp authentication ms-chap
!
```

```
ip classless
ip route 172.29.1.129 255.255.255.255 1.1.1.1
ip route 172.29.63.9 255.255.255.255 1.1.1.1
no ip http server
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

VPN Tunnel Authentication Examples

The following examples shows several possibilities for performing local tunnel authentication. These examples only show the information relevant to tunnel authentication.

Tunnel Secret Configured Using the Local Name Command

The following examples are for a NAS and tunnel server that configure the tunnel names by using **local name** VPN group commands. The NAS tunnel name is ISP_NAS, the tunnel server tunnel name is ENT_HGW, and the tunnel secret is tunnelme.

NAS Configuration

The NAS tunnel name is specified by the **local name** command. The tunnel server tunnel name and tunnel secret are specified by the **username** command.

```
username ENT_HGW password 7 tunnelme
.
.
.
vpdn-group 1
 local name ISP_NAS
```

Tunnel Server Configuration

The tunnel server tunnel name is specified by the **local name** command. The NAS tunnel name and tunnel secret are specified by the **username** command.

```
username ISP_NAS password 7 tunnelme
.
.
.
vpdn-group 1
 local name ENT_HGW
```

Tunnel Secret Configured Using the L2TP Tunnel Password Command

The following example is for a NAS and tunnel server that both configure the tunnel secret using the **l2tp tunnel password** command. Because both routers use this command, they do not need to use either **username** or **local name** commands for tunnel authentication. The tunnel secret is tunnelme.

NAS Configuration

```
vpdn-group 1
 request-dialin
  protocol l2tp
 l2tp tunnel password tunnelme
```

Tunnel Server Configuration

```
vpdn-group 1
 accept-dialin
  protocol l2tp
 l2tp tunnel password tunnelme
```

Tunnel Secret Configuration Using Different Tunnel Authentication Methods

The follow example is for a NAS that uses the **username** command to specify the tunnel secret and a tunnel server that uses the **l2tp tunnel password** command to specify the tunnel secret.

NAS Configuration

```
username adrian password garf1eld
.
.
.
vpdn-group 1
 local name stella
```

Tunnel Server Configuration

```
vpdn-group 1
 accept--dialin
  protocol l2tp
 local name adrian
 l2tp tunnel password garf1eld
```

NAS Comprehensive Dial-In Configuration Example

The following example shows a NAS configured to tunnel PPP calls to a tunnel server using L2TP and local authentication and authorization:

```
! Enable AAA authentication and authorization with RADIUS as the default method
aaa new-model
aaa authentication ppp default radius
aaa authorization network default radius
!
username ISP_NAS password 7 tunnelme
username ENT_HGW password 7 tunnelme
!
vpdn enable
!
! Configure VPN to first search on the client domain name and then on the DNIS
vpdn search-order domain dnis
! Allow a maximum of 10 simultaneous VPN sessions
vpdn session-limit 10
!
! Configure VPN to initiate VPN dial-in sessions
vpdn-group 1
 request-dialin
! Specify L2TP as the tunneling protocol
  protocol l2TP
! Tunnel clients with the domain name "hgw.com"
  domain hgw.com
! Establish a tunnel with IP address 172.22.66.25
 initiate-to ip 172.22.66.25
! Identify the tunnel using the name "ISP_NAS"
 local name ISP_NAS
!
```

```
! Defines the ISDN switch type as primary-5ess
isdn switch-type primary-5ess
!
! Commissions the T1 controller to allow modem calls in to the NAS
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Ethernet0
 ip address 172.22.66.23 255.255.255.192
!
! Configure the Serial channel to allow modem calls in to the NAS
interface Serial0:23
 no ip address
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no cdp enable
!
!
interface Group-Async1
 ip unnumbered Ethernet0
 encapsulation ppp
 async mode interactive
 no peer default ip address
 ppp authentication chap pap
group-range 1 96
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.22.66.1
!
! Specifies the RADIUS server IP address, authorization port, and accounting port
radius-server host 172.22.66.16 auth-port 1645 acct-port 1646
! Specifies the authentication key to be used with the RADIUS server
radius-server key cisco
!
line con 0
 transport input none
! Configures the modems
line 1 96
 autoselect during-login
 autoselect ppp
 modem InOut
line aux 0
line vty 0 4
!
end
```

Tunnel Server Comprehensive Dial-in Configuration Example

The following example show a tunnel server configured to accept L2TP tunnels from a NAS using local authentication and authorization:

```
aaa new-model
! Configure AAA to first use the local database and then contact the RADIUS server for
! PPP authentication
aaa authentication ppp default local radius
! Configure AAA network authorization and accounting by using the RADIUS server
aaa authorization network default radius
aaa accounting network default start-stop radius
!
```

```
username ISP_NAS password 7 tunnelme
username ENT_HGW password 7 tunnelme
!
vpdn enable
! Prevent any new VPN sessions from being established without disturbing existing
! sessions
vpdn softshut

!
! Configure VPN to accept dial-in sessions
vpdn-group 1
 accept-dialin
! Specify L2TP as the tunneling protocol
  protocol l2tp
! Specify that virtual-access interfaces be cloned from virtual template 1
  virtual-template 1
! Accept dial-in requests from a router using the tunnel name "ISP_NAS"
 terminate-from hostname ISP_NAS
! Identify the tunnel using the tunnel name "ENT_HGW"
 local name ENT_HGW
!
interface Ethernet0/0
 ip address 172.22.66.25 255.255.255.192
 no ip directed-broadcast
!
interface Virtual-Template1
! Use the IP address of interface Ethernet 0
 ip unnumbered Ethernet0
! Returns an IP address from the default pool to the VPN client
peer default ip address pool default
! Use CHAP to authenticate PPP
 ppp authentication chap
!
ip local pool default 172.30.2.1 172.30.2.96
ip classless
ip route 0.0.0.0 0.0.0.0 172.22.66.1
!
! Specifies the RADIUS server IP address, authorization port, and accounting port
radius-server host 172.22.66.13 auth-port 1645 acct-port 1646
! Specifies the authentication key to be used with the RADIUS server
radius-server key cisco
```

NAS Configured for Both Dial-In and Dial-Out Example

You can configure a NAS to simultaneously initiate L2TP or L2F dial-in tunnels to a tunnel server and also accept L2TP dial-out tunnels from a tunnel server.

In the following example, the VPN group of a NAS is configured to dial in using L2F and to dial out using L2TP as the tunneling protocol and dialer interface 2. The example only shows the VPN group and dialer configuration:

```
vpdn-group 1
 request-dialin
  protocol l2f
  domain jgb.com
 accept-dialout
  protocol l2tp
  dialer 2
 local name cerise
 terminate-from hostname reuben
 initiate-to ip 172.1.2.3
!
```

```
interface Dialer2
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer in-band
 dialer aaa
 dialer-group 1
 ppp authentication chap
```

Tunnel Server Configured for Both Dial-In and Dial-Out Example

You can configure a tunnel server to simultaneously receive L2TP or L2F dial-in tunnels from a NAS and also initiate L2TP dial-out tunnels to a NAS.

In the following example, a tunnel server VPN group is configured to dial in using virtual template 1 to clone the virtual access interface and to dial out using dialer pool 1. The example only shows the VPN group and dialer configuration:

```
vpdn-group 1
 accept-dialin
  protocol l2tp
  virtual-template 1
 request-dialout
  protocol l2tp
  pool-member 1
 local name reuben
 terminate-from hostname cerise
 initiate-to ip 10.3.2.1
!
interface Dialer2
 ip address 172.19.2.3 255.255.255.128
 encapsulation ppp
 dialer remote-name reuben
 dialer string 5551234
 dialer vpdn
 dialer pool 1
 dialer-group 1
 ppp authentication chap
```

RADIUS Profile Examples

The following sections show VPN RADIUS profiles configured using CiscoSecure version 2.3.1:

- RADIUS Domain Profile
- RADIUS User Profile

RADIUS Domain Profile

The following example show a profile that is configured on the NAS RADIUS server to tunnel calls from users who dial-in with the domain name terrapin.com. The NAS will balance calls between the tunnel servers at 172.16.171.11 and 172.16.171.12. If both of those tunnel servers are unavailable, the NAS will tunnel calls to 172.16.171.13.

```
user = terrapin.com{
profile_id = 29
set server current-failed-logins = 0
profile_cycle = 7
radius=Cisco {
```

```
check_items= {
2=cisco
}
reply_attributes= {
9,1="vpdn:l2tp-tunnel-password=cisco123"
9,1="vpdn:tunnel-type=l2tp"
9,1="vpdn:ip-addresses=172.16.171.11 172.16.171.12/172.16.171.13"
9,1="vpdn:tunnel-id=tunnel"
}
}
}
```

Note check_items={2=cisco} is a hard-coded password. This password must be "cisco."

RADIUS User Profile

The following example shows a profile that is configured on the tunnel server RADIUS server to authorize and authenticate user sailor@terrapin.com:

```
user = sailor@terrapin.com{
profile_id = 28
profile_cycle = 2
radius=Cisco {
check_items= {
2=cisco
}
reply_attributes= {
6=2
7=1
}
}
}
```

Note check_items={2=cisco} is a hard-coded password. This password must be "cisco."

TACACS+ Profile Examples

The following sections show VPN TACACS+ profiles configured using CiscoSecure version 2.2.2:

- TACACS+ Domain Profile
- TACACS+ User Profile
- TACACS+ Tunnel Profiles

TACACS+ Domain Profile

The following example shows a profile that is configured on the NAS TACACS+ server to tunnel users who dial in with the domain name guava.com:

```
user = guava.com{
profile_id = 83
profile_cycle = 1
service=ppp {
protocol=vpdn {
set tunnel-id=isp
set ip-addresses="10.31.1.50"
```

```
set nas-password="little"
set gw-password="birdies"
}
protocol=lcp {
}
}
}
```

TACACS+ User Profile

The following example shows a profile that is configured on the tunnel server TACACS+ to authorize and authenticate user geaner@guava.com:

```
user = geaner@guava.com{
profile_id = 85
profile_cycle = 1
password = chap "daisies"
service=ppp {
protocol=ip {
default attribute=permit
}
protocol=lcp {
}
}
}
```

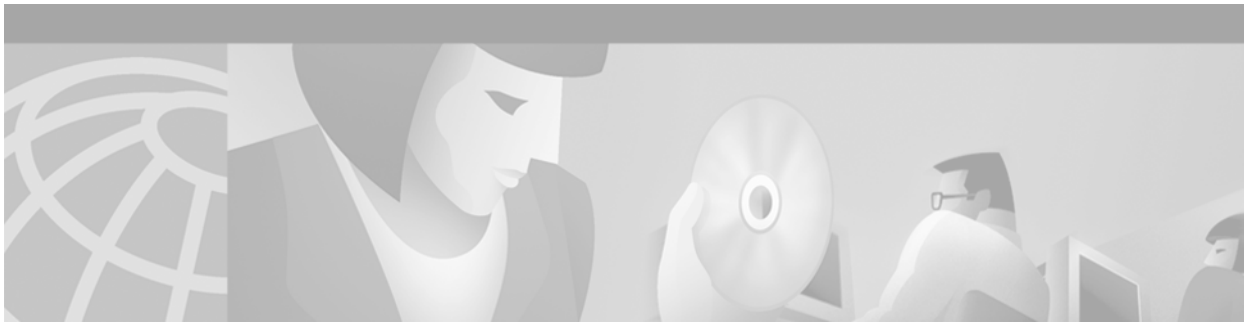
TACACS+ Tunnel Profiles

The following examples show a profile that is configured on the tunnel server TACACS+ server to authenticate the tunnel. See the "Configuring VPN Tunnel Authentication Using the Host Name or Local Name" and "Configuring VPN Tunnel Authentication Using the L2TP Tunnel Password" sections earlier in this chapter for more information on tunnel authentication.

Note Only the tunnel server AAA server can perform tunnel authentication. Tunnel authentication must be performed locally by the NAS.

```
user = tunnel-server {
profile_id = 82
profile_cycle = 1
password = chap "3stone"
service=ppp {
protocol=ip {
default attribute=permit
}
protocol=lcp {
}
}
}
```

PPP Configuration



Configuring Asynchronous SLIP and PPP

This chapter describes how to configure asynchronous Serial Line Internet Protocol (SLIP) and PPP. It includes the following main sections:

- Asynchronous SLIP and PPP Overview
- How to Configure Asynchronous SLIP and PPP
- Configuration Examples for Asynchronous SLIP and PPP

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Asynchronous SLIP and PPP Overview

PPP and SLIP define methods of sending IP packets over standard asynchronous serial lines with minimum line speeds of 1200 baud.

Using SLIP or PPP encapsulation over asynchronous lines is an inexpensive way to connect personal computers (PCs) to a network. PPP and SLIP over asynchronous dialup modems allow a home computer to be connected to a network without the cost of a leased line. Dialup PPP and SLIP links can also be used for remote sites that need only occasional remote node or backup connectivity. Both public-domain and vendor-supported PPP and SLIP implementations are available for a variety of computer applications.

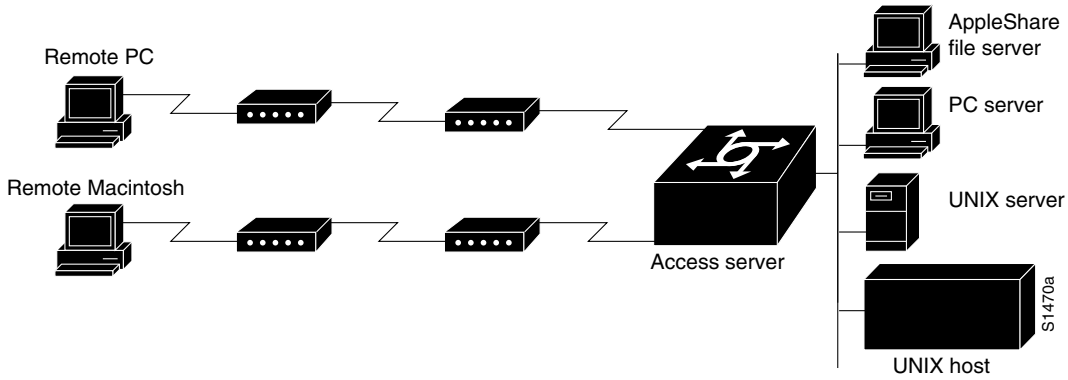
The Cisco IOS software concentrates a large number of SLIP or PPP PC or workstation client hosts onto a network interface that allows the PCs to communicate with any host on the network. The Cisco IOS software can support any combination of SLIP or PPP lines and lines dedicated to normal asynchronous devices such as terminals and modems. Refer to RFC 1055 for more information about SLIP, and RFCs 1331 and 1332 for more information about PPP.

SLIP is an older protocol. PPP is a newer, more robust protocol than SLIP, and it contains functions that can detect or prevent misconfiguration. PPP also provides greater built-in security mechanisms.

Note Most asynchronous serial links have very low bandwidth. Take care to configure your system so the links will not be overloaded. Consider using default routes and filtering routing updates to prevent them from being sent on these asynchronous lines.

Figure 81 illustrates a typical asynchronous SLIP or PPP remote-node configuration.

Figure 81 Sample SLIP or PPP Remote-Node Configuration



Responding to BOOTP Requests

The BOOTP protocol allows a client machine to discover its own IP address, the address of the router, and the name of a file to be loaded in to memory and executed. There are typically two phases to using BOOTP: first, the client's address is determined and the boot file is selected; then the file is transferred, typically using the TFTP.

PPP and SLIP clients can send BOOTP requests to the Cisco IOS software, and the Cisco IOS software responds with information about the network. For example, the client can send a BOOTP request to learn its IP address and where the boot file is located, and the Cisco IOS software responds with the information.

BOOTP supports the extended BOOTP requests specified in RFC 1084 and works for both PPP and SLIP encapsulation.

BOOTP compares to Reverse Address Resolution Protocol (RARP) as follows: RARP is an older protocol that allows a client to determine its IP address if it knows its hardware address. (Refer to the *Cisco IOS IP Configuration Guide* for more information about RARP.) However, RARP is a hardware link protocol, so it can be implemented only on hosts that have special kernel or driver modifications that allow access to these raw packets. BOOTP does not require kernel modifications.

Asynchronous Network Connections and Routing

Line configuration commands configure a connection to a terminal or a modem. Interface configuration (**async**) commands, described in this chapter, configure a line as an asynchronous network interface over which networking functions are performed.

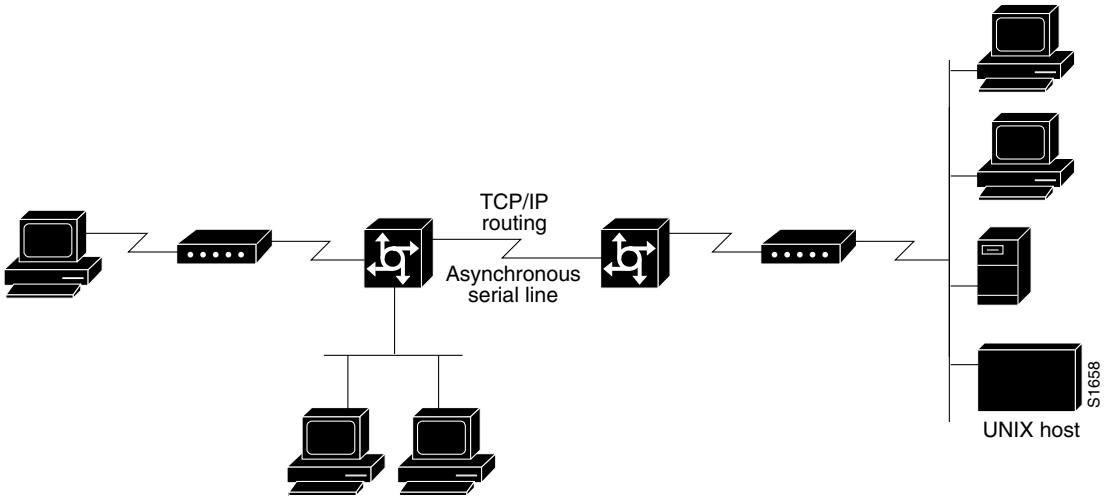
The Cisco IOS software also supports IP routing connections for communication that requires connecting one network to another.

The Cisco IOS software supports protocol translation for PPP and SLIP between other network devices running Telnet, local-area transport (LAT), or X.25. For example, you can send IP packets across a public X.25 packet assembler/disassembler (PAD) network using SLIP or PPP encapsulation when SLIP or PPP protocol translation is enabled. For more information, see the chapter "Configuring Protocol Translation and Virtual Asynchronous Devices" in this publication.

If asynchronous dynamic routing is enabled, you can enable routing at the user level by using the **routing** keyword with the **slip** or **ppp** EXEC command.

Asynchronous interfaces offer both dedicated and dynamic address assignment, configurable hold queues and IP packet sizes, extended BOOTP requests, and permit and deny conditions for controlling access to lines. Figure 82 shows a sample asynchronous routing configuration.

Figure 82 Sample Asynchronous Routing Configuration



Asynchronous Interfaces and Broadcasts

The Cisco IOS software recognizes a variety of IP broadcast addresses. When a router receives an IP packet from an asynchronous client, it rebroadcasts the packet onto the network without changing the IP header.

The Cisco IOS software receives the SLIP or PPP client broadcasts and responds to BOOTP requests with the current IP address assigned to the asynchronous interface from which the request was received. This facility allows the asynchronous client software to automatically learn its own IP address.

How to Configure Asynchronous SLIP and PPP

To configure SLIP and PPP, perform the tasks in the following sections; all tasks are optional:

- Configuring Network-Layer Protocols over PPP and SLIP (Optional)
- Configuring Asynchronous Host Mobility (Optional)
- Making Additional Remote Node Connections (Optional)

- Configuring Remote Access to NetBEUI Services (Optional)
- Configuring Performance Parameters (Optional)

Configuring Network-Layer Protocols over PPP and SLIP

You can configure network-layer protocols, such as AppleTalk, IP, and Internet Protocol Exchange (IPX), over PPP and SLIP. SLIP supports only IP, but PPP supports each of these protocols. See the sections that follow to configure these protocols over PPP and SLIP.

Configuring IP and PPP

To enable IP-PPP (IPCP) on a synchronous or asynchronous interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip address` *ip-address mask* `[secondary]`	Configures IP routing on the interface.
	or	
	`Router(config-if)# ip unnumbered` *type number*	Configures IP unnumbered routing on a serial interface.
Step 2	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the serial interface.
Step 3	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.

Configuring IPX and PPP

You can configure IPX over PPP (IPXCP) on synchronous serial and asynchronous serial interfaces using one of two methods.

The first method associates an asynchronous interface with a loopback interface configured to run IPX. It permits you to configure IPX-PPP on asynchronous interfaces only.

The second method permits you to configure IPX-PPP on asynchronous and synchronous serial interfaces. However, it requires that you specify a dedicated IPX network number for each interface, which can require a substantial number of network numbers for a large number of interfaces.

You can also configure IPX to run on virtual terminal lines configured for PPP. See the section "Enabling IPX and PPP over X.25 to an IPX Network on Virtual Terminal Lines" later in this chapter.

Note If you are configuring IPX-PPP on asynchronous interfaces, you should filter routing updates on the interface. Most asynchronous serial links have very low bandwidth, and routing updates take up a great deal of bandwidth. The previous task table uses the **ipx update interval** command to filter SAP updates. For more information about filtering routing updates, see the section about creating filters for updating the routing table in the chapter "Configuring Novell IPX" in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

IPX and PPP and Associating Asynchronous Interfaces with Loopback Interfaces

To permit IPX client connections to an asynchronous interface, the interface must be associated with a loopback interface configured to run IPX. To permit such connections, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx routing [node]`	Enables IPX routing.
Step 2	`Router(config)# interface loopback number`	Creates a loopback interface, which is a virtual interface existing only inside the router, and begins interface configuration mode.
Step 3	`Router(config-if)# ipx network network`[1]	Enables IPX routing on the loopback interface.
Step 4	`Router(config-if)# exit`	Exits to global configuration mode.
Step 5	`Router(config)# interface async number`	Enters interface configuration mode for the asynchronous interface.
Step 6	`Router(config-if)# ip unnumbered type number`	Configures IP unnumbered routing on the interface.
Step 7	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 8	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.
Step 9	`Router(config-if)# ipx ppp-client loopback number`	Assigns the asynchronous interface to the loopback interface configured for IPX.
Step 10	`Router(config-if)# ipx update interval`	Turns off Service Advertising Protocol (SAP) updates to optimize bandwidth on asynchronous interfaces.

1. Every interface must have a unique IPX network number.

IPX and PPP Using Dedicated IPX Network Numbers for Each Interface

To enable IPX and PPP, use the following commands beginning in global configuration mode. The first five steps are required. The last step is optional.

	Command	Purpose
Step 1	`Router(config)# ipx routing [node]`	Enables IPX routing.
Step 2	`Router(config)# interface loopback number`	Creates a loopback interface, which is a virtual interface existing only inside the router, and begins interface configuration mode.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 4	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.
Step 5	`Router(config-if)# ipx network network`[1]	Enables IPX routing on the interface.
Step 6	`Router(config-if)# ipx update interval`	(Optional) Turns off SAP updates to optimize bandwidth on asynchronous interfaces.

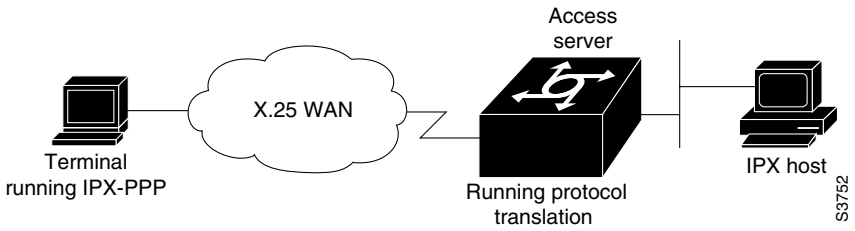
1. Every interface must have a unique IPX network number.

Enabling IPX and PPP over X.25 to an IPX Network on Virtual Terminal Lines

You can enable IPX-PPP on virtual terminal lines, which permits clients to log in to a virtual terminal on a router, invoke a PPP session at the EXEC prompt to a host, and run IPX to the host.

For example, in Figure 83, the client terminal on the X.25 network logs in to the access server via a virtual terminal line, which is configured for IPX-PPP. When the user connects to the access server and the EXEC prompt appears, enter the PPP command to connect to the IPX host. The virtual terminal is configured to run IPX, so when the PPP session is established from the access server, the terminal can access the IPX host using an IPX application.

Figure 83 IPX-PPP on a Virtual Asynchronous Interface



To enable IPX to run over your PPP sessions on virtual terminal lines, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx routing [node]`	Enables IPX routing.
Step 2	`Router(config)# interface loopback number`	Creates a loopback interface and begins interface configuration mode.
Step 3	`Router(config-if)# ipx network network`[1]	Enables a virtual IPX network on the loopback interface.
Step 4	`Router(config-if)# vty-async ipx ppp-client loopback number`	Enables IPX-PPP on virtual terminal lines by assigning it to the loopback interface configured for IPX.

1. Every loopback interface must have a unique IPX network number.

Configuring AppleTalk and PPP

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

You create a virtual network that exists only for accessing an AppleTalk internet through the server. To create a new AppleTalk zone, enter the **appletalk virtual-net** command and use a new zone name; this network number is then the only one associated with this zone. To add network numbers to an existing AppleTalk zone, use this existing zone name in the command; this network number is then added to the existing zone. Routing is not supported on these interfaces.

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

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Defines encapsulation as PPP on this interface.
Step 2	`Router(config-if)# appletalk virtual-net network-number zone-name`	Creates an internal network on the server.
Step 3	`Router(config-if)# appletalk client-mode`	Enables client-mode on this interface.

Configuring IP and SLIP

To enable IP-SLIP on a synchronous or asynchronous interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip address ip-address mask`	Configures IP routing on the interface.
	or	
	`Router(config-if)# ip unnumbered type number`	Configures IP unnumbered routing on a serial interface.
Step 2	`Router(config-if)# encapsulation slip`	Enables SLIP encapsulation on the serial interface.
Step 3	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.

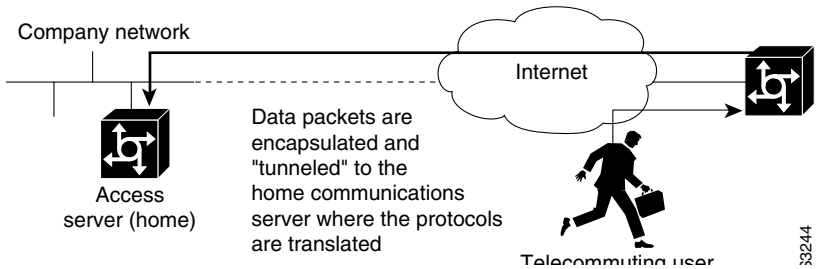
Configuring Asynchronous Host Mobility

The access server supports a packet tunneling strategy that extends the internetwork—in effect creating a virtual private link for the mobile user. When a user activates asynchronous host mobility, the access server on which the remote user dials in becomes a remote point of presence (POP) for the home network of the user. Once logged in, users experience a server environment identical to the one that they experience when they connect directly to the "home" access server.

Once the network-layer connection is made, data packets are tunneled at the physical or data link layer instead of at the protocol layer. In this way, raw data bytes from dial-in users are transported directly to the "home" access server, which processes the protocols.

Figure 84 illustrates the implementation of asynchronous host mobility on an extended internetwork. A mobile user connects to an access server on the internetwork and, by activating asynchronous host mobility, is connected to a "home" access server configured with the appropriate username. The user sees an authentication dialog or prompt from the "home" system and can proceed as if he or she were connected directly to that device.

Figure 84 Asynchronous Host Mobility



Asynchronous host mobility is enabled with the **tunnel** EXEC command and the **ip tcp async-mobility server** global configuration command. The **ip tcp async-mobility server** command establishes asynchronous listening on TCP tunnel port 57. The **tunnel** command sets up a network-layer connection to the specified destination. Both commands must be used. The access server accepts the connection, attaches it to a virtual terminal line, and runs a command parser capable of running the normal dial-in services. After the connection is established, data is transferred between the modem and network connection with a minimum of interpretations. When communications are complete, the network connection can be closed and terminated from either end.

To enable asynchronous host mobility, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip tcp async-mobility server`	Enables asynchronous listening on TCP tunnel port 57.
Step 2	`Router(config)# exit`	Returns to user EXEC mode.
Step 3	`Router# tunnel host`	Sets up a network-layer connection to a router by specifying its Internet name or address. Replace the *host* argument with the name or address of the device that you want to connect to.

To connect from a router other than a Cisco router, you must use Telnet. After a connection is established, you receive an authentication dialog or prompt from your home router, and can proceed as if you are connected directly to that router. When communications are complete, the network connection can be closed and terminated from either end of the connection.

Making Additional Remote Node Connections

This section describes how to connect devices across telephone lines by using PPP and SLIP. It includes the following sections:

- Creating PPP Connections
- Making SLIP Connections

Creating PPP Connections

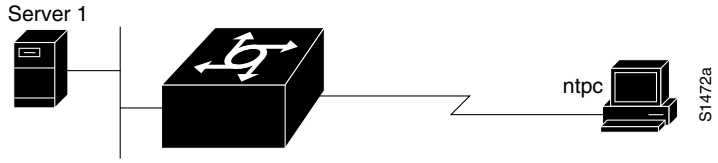
When you connect from a remote node computer through an asynchronous port on an access server to the EXEC facility to connect from the access server to a device on the network, use the following command in EXEC mode:

Command	Purpose
`Router> ppp {/default | {remote-ip-address | remote-name} [@tacacs-server]} [/routing]`	Creates a PPP connection.

If you specify an address for the TACACS server using **/default** or *tacacs-server*, the address must be the first parameter in the command after you type **ppp**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

For example, if you are working at home on the device named ntpc in Figure 85 and want to connect to Server 1 using PPP, you could dial in to the access server. When you connect to the EXEC prompt on the access server, enter the **ppp** command to connect with the device.

Figure 85 Using the ppp Command



To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from EXEC mode by using the **exit** command.

Making SLIP Connections

To make a serial connection to a remote host by using SLIP, use the following command in EXEC mode:

Command	Purpose
`Router> slip [/default] {remote-ip-address | remote-name} [@tacacs-server] [/routing]} [/compressed]`	Creates a SLIP connection.

Your system administrator can configure SLIP to expect a specific address or to provide one for you. It is also possible to set up SLIP in a mode that compresses packets for more efficient use of bandwidth on the line.

If you specify an address for the TACACS server using **/default** or *tacacs-server*, the address must be the first parameter in the command after you type **slip**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

If you do not use the *tacacs-server* argument to specify a TACACS server for SLIP address authentication, the TACACS server specified at login (if any) is used for the SLIP address query.

To optimize bandwidth on a line, SLIP enables compression of the SLIP packets using Van Jacobson TCP header compression as defined in RFC 1144.

To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from EXEC mode by using the **exit** command.

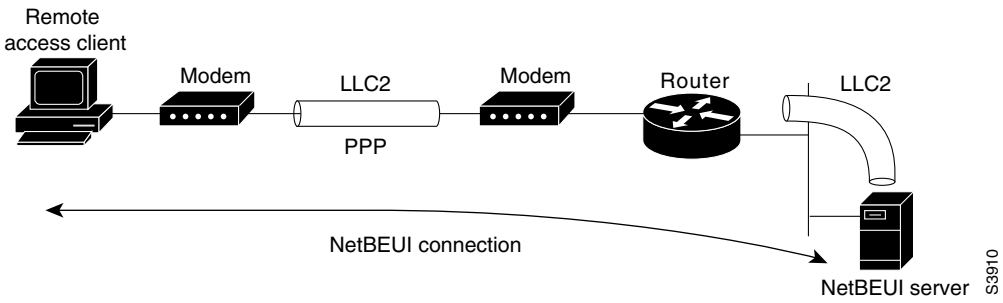
Configuring Remote Access to NetBEUI Services

NetBIOS Extended User Interface (NetBEUI) is a simple networking protocol developed by IBM for use by PCs in a LAN environment. It is an extension of the original Network Basic Input/Output System (NetBIOS) from IBM. NetBEUI uses a broadcast-based name to 802.x address translation mechanism. Because NetBEUI has no network layer, it is a nonroutable protocol.

The NetBIOS Frames Control Protocol (NBFCP) enables packets from a NetBEUI application to be transferred via a PPP connection. NetBEUI/PPP is supported in the access server and Cisco enterprise images only.

Using the Cisco IOS implementation, remote NetBEUI users can have access to LAN-based NetBEUI services. The PPP link becomes the ramp for the remote node to access NetBIOS services on the LAN. (See Figure 86.) An Logical Link Control, type 2 (LLC2) connection is set up between the remote access client and router, and a second LLC2 connection is set up between the router and the remote access (NetBEUI) server.

Figure 86 NetBEUI Connection



By supporting NetBEUI remote clients over PPP, Cisco routers function as a native NetBEUI dial-in router for remote NetBEUI clients. Thus, you can offer remote access to a NetBEUI network through asynchronous or ISDN connections.

To enable a remote access client using a NetBEUI application to connect with the remote router providing NetBEUI services, configure interfaces on the remote access client side and the remote router side by using the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# netbios nbf`	Enables NBFCP on each side of a NetBEUI connection.

To view NetBEUI connection information, use the following command in EXEC mode:

Command	Purpose
`Router> show nbf sessions`	Views NetBEUI connection information.

Configuring Performance Parameters

To tune IP performance, complete the tasks in the following sections:

- Compressing TCP Packet Headers (As required)
- Setting the TCP Connection Attempt Time (As required)
- Compressing IPX Packet Headers over PPP (As required)
- Enabling Fast Switching (As required)
- Controlling Route Cache Invalidation (As required)
- Customizing SLIP and PPP Banner Messages (As required)

Compressing TCP Packet Headers

You can compress the headers of your TCP/IP packets to reduce their size and thereby increase performance. Header compression is particularly useful on networks with a large percentage of small packets, such as those supporting many Telnet connections. This feature compresses only the TCP

header, so it has no effect on UDP packets or other protocol headers. The TCP header compression technique, described fully in RFC 1144, is supported on serial lines using High-Level Data Link Control (HDLC) or PPP encapsulation. You must enable compression on both ends of a serial connection.

You can optionally specify outgoing packets to be compressed only when TCP incoming packets on the same interface are compressed. If you do not specify this option, the Cisco IOS software will compress all traffic. The default is no compression.

You can also specify the total number of header compression connections that can exist on an interface. You should configure one connection for each TCP connection through the specified interface.

To enable compression, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip tcp header-compression [passive]`	Enables TCP header compression.
Step 2	`Router(config-if)# ip tcp compression-connections` *number*	Specifies the total number of header compression connections that can exist on an interface.

Note When compression is enabled, fast switching is disabled. Fast processors can handle several fast interfaces, such as T1 lines, that are running header compression. However, you should think carefully about traffic characteristics in your network before compressing TCP headers. You might want to use the monitoring commands to help compare network utilization before and after enabling header compression.

Setting the TCP Connection Attempt Time

You can set the amount of time that the Cisco IOS software will wait to attempt to establish a TCP connection. In previous versions of the Cisco IOS software, the system would wait a fixed 30 seconds when attempting to make the connection. This amount of time is not enough in networks that have dialup asynchronous connections, such as a network consisting of dial-on-demand links that are implemented over modems, because it will affect your ability to use Telnet over the link (from the router) if the link must be brought up.

Because the connection attempt time is a host parameter, it does not pertain to traffic going through the router, just to traffic originated at it.

To set the TCP connection attempt time, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp synwait-time` *seconds*	Sets the amount of time for which the Cisco IOS software will wait to attempt to establish a TCP connection.

Compressing IPX Packet Headers over PPP

The Cisco IOS software permits compression of IPX packet headers over various WAN media. There are two protocols for IPX compression on point-to-point links:

- CIPX, also known as Telebit style compression
- Shiva compression, which is proprietary

Cisco routers support IPX Header Compression (CIPX) on all point-to-point Novell interfaces over various WAN media.

CIPX is described in RFC 1553, *Compressing IPX Headers Over WAN Media*. The CIPX algorithm is based on the same concepts as Van Jacobson TCP/IP header compression algorithm. CIPX operates over PPP WAN links using either the IPXCP or IPXWAN communications protocols.

CIPX compresses all IPX headers and IPX/NCP headers for Novell packets with the following Network Control Program (NCP) packet types:

- 0x2222—NCP request from workstation
- 0x3333—NCP replies from file server

In this version of software, CIPX is configurable only for PPP links.

CIPX header compression can reduce header information from 30 bytes down to as little as 1 byte. This reduction can save bandwidth and reduce costs associated with IPX routing over WAN links that are configured to use IPXCP or IPXWAN.

Consider the following issues before implementing CIPX:

- CIPX is supported on all point-to-point IPX interfaces using PPP or IPXWAN processing (or both).
- CIPX needs to be negotiated for both directions of the link, because it uses the reverse direction of the link for communicating decompression problems back to the originating peer. In other words, all peer routers must have CIPX enabled.

To configure CIPX, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx compression cipx` *number-of-slots*	Compresses IPX packet headers in a PPP session.

Note We recommend that you keep a slot value of 16. Because slots are maintained in the router buffer, a larger number can impact buffer space for other operations.

Enabling Fast Switching

Fast switching involves the use of a high-speed switching cache for IP routing. With fast switching, destination IP addresses are stored in the high-speed cache so that some time-consuming table lookups can be avoided. The Cisco IOS software generally offers better packet transfer performance when fast switching is enabled.

To enable or disable fast switching, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip route-cache`	Enables fast-switching (use of a high-speed route cache for IP routing).
Step 2	`Router(config-if)# no ip route-cache`	Disables fast switching and enables load balancing on a per-packet basis.

Controlling Route Cache Invalidation

The high-speed route cache used by IP fast switching is invalidated when the IP routing table changes. By default, the invalidation of the cache is delayed slightly to avoid excessive CPU load while the routing table is changing.

To control route cache invalidation, use the following commands in global configuration mode as needed for your network:

Note This task normally should not be necessary. It should be performed only under the guidance of technical staff. Incorrect configuration can seriously degrade the performance of your router.

	Command	Purpose
Step 1	`Router(config)# no ip cache-invalidate-delay`	Allows immediate invalidation of the cache.
Step 2	`Router(config)# ip cache-invalidate-delay [minimum maximum quiet-threshold]`	Delays invalidation of the cache.

Customizing SLIP and PPP Banner Messages

This feature enables you to customize the banner that is displayed when making a SLIP or PPP connection to avoid connectivity problems the default banner message causes in some non-Cisco SLIP and PPP dialup software. This feature is particularly useful when legacy client applications require a specialized connection string.

To configure the SLIP-PPP banner message, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner slip-ppp d message d`	Configures the SLIP-PPP banner to display a customized message.

You can also use tokens in the banner message to display current IOS configuration variables. Tokens are keywords of the form **$(***token***)**. When you include tokens in a banner command, Cisco IOS will replace **$(***token***)** with the corresponding configuration variable.

Table 35 lists the tokens that you can use in the **banner slip-ppp** command.

Table 35 SLIP Banner Tokens

Tokens	Information Displayed in Banner
Global	
$(hostname)	Hostname of the router
$(domain)	Domain name of the router
Slip/PPP Banner-Specific	
$(peer-ip)	IP address of the peer machine
$(gate-ip)	IP address of the gateway machine
$(encap)	Encapsulation type (SLIP, PPP, and so on)

Table 35 SLIP Banner Tokens (continued)

$(encap-alt)	Encapsulation type displayed as SL/IP instead of SLIP
$(mtu)	MTU size

Configuration Examples for Asynchronous SLIP and PPP

This section provides the following examples:

- Basic PPP Configurations Examples
- Remote Node NetBEUI Examples
- Remote Network Access Using PPP Basic Configuration Example
- Remote Network Access Using PPP and Routing IP Example
- Remote Network Access Using a Leased Line with Dial-Backup and PPP Example
- Multilink PPP Using Multiple Asynchronous Interfaces Example

Basic PPP Configurations Examples

The following example illustrates how to make a connection when the system administrator defines a default IP address by including the **peer default ip address** command in interface configuration mode.

Note The **peer default ip address** command replaces the **async default ip address** command.

Once a correct password is entered, you are placed in SLIP mode, and the IP address appears:

```
Router> slip
Password:
Entering SLIP mode.
Your IP address is 192.168.7.28, MTU is 1524 bytes
```

The following example shows the prompts displayed and the response required when dynamic addressing is used to assign the SLIP address:

```
Router> slip
IP address or hostname? 192.168.6.15
Password:
Entering SLIP mode
Your IP address is 192.168.6.15, MTU is 1524 bytes
```

In the previous example, the address 192.168.6.15 had been assigned as the default. Password verification is still required before SLIP mode can be enabled, as follows:

```
Router> slip default
Password:
Entering SLIP mode
Your IP address is 192.168.6.15, MTU is 1524 bytes
```

The following example illustrates the implementation of header compression on the interface with the IP address 172.16.2.1:

```
Router> slip 172.16.2.1 /compressed
Password:
```

```
Entering SLIP mode.
Interface IP address is 172.16.2.1, MTU is 1500 bytes.
Header compression will match your system.
```

In the preceding example, the interface is configured for **ip tcp header-compression passive**, which permitted the user to enter the **/compressed** keyword at the EXEC mode prompt. The message "Header compression will match your system" indicates that the user has specified compression. If the line was configured for **ip tcp header-compression on**, this line would read "Header compression is On."

The following example specifies a TACACS server named parlance for address authentication:

```
Router> slip 10.0.0.1@parlance
Password:
Entering SLIP mode.
Interface IP address is 10.0.0.1, MTU is 1500 bytes
Header compression will match your system.
```

The following example sets the SLIP-PPP banner using several tokens and the percent sign (%) as the delimiting character:

```
Router(config)# banner slip-ppp %
Enter TEXT message.  End with the character '%'.
Starting $(encap) connection from $(gate-ip) to $(peer-ip) using a maximum packet size of
$(mtu) bytes... %
```

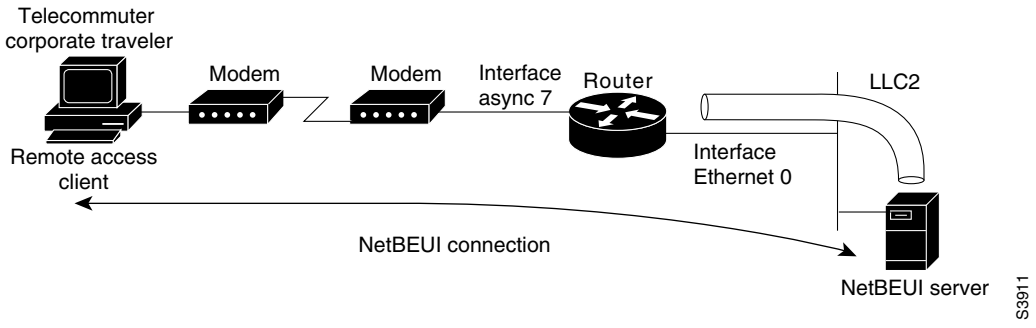
When you enter the **slip** command, you will see the following banner. Notice that the **$(***token***)** syntax is replaced by the corresponding configuration variables.

```
Starting SLIP connection from 192.168.69.96 to 172.16.80.8 using a maximum packet size of
1500 bytes...
```

Remote Node NetBEUI Examples

In the following example, asynchronous interface 7 and Ethernet interface 0 are configured to enable NetBEUI connectivity between the corporate telecommuter client and the remote access (NetBEUI) server. The PC client is running the Chat legacy application in Windows NT to connect with the remote server. (See Figure 87.)

Figure 87 Connecting a Remote NetBEUI Client to a Server Through a Router



The configuration for the router is as follows:

```
interface async 7
 netbios nbf
 encapsulation ppp
```

You would also need to configure security, such as TACACS+, RADIUS, or another form of login authentication on the router.

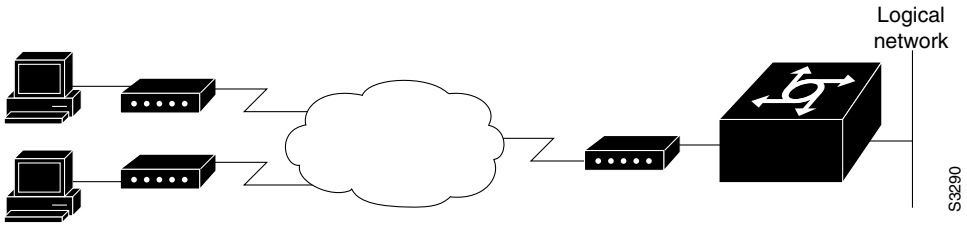
Remote Network Access Using PPP Basic Configuration Example

Figure 88 illustrates a simple network configuration that includes remote PCs with modems connected via modem to a router. The cloud is a Public Switched Telephone Network (PSTN). The modems are connected via asynchronous lines, and the access server is connected to a local network.

In this example, the following is configured:

- An asynchronous line on the access server configured to use PPP encapsulation.
- An interface on the access server for the modem connection; this interface also needs to be configured to accept incoming modem calls.
- A default IP address for each incoming line.

Figure 88 Remote Network Access Using PPP



This default address indicates the address of the remote PC to the server, unless the user explicitly specifies another when starting the PPP session.

The server is configured for interactive mode with autoselect enabled, which allows the user to automatically begin a PPP session upon detection of a PPP packet from the remote PC; or, the remote PC can explicitly begin a PPP session by entering the **ppp** EXEC command at the prompt.

The configuration is as follows:

```
ip routing
!
interface ethernet 0
 ip address 192.168.32.12 255.255.255.0
!
interface async 1
 encapsulation ppp
 async mode interactive
 async default ip address 192.168.32.51
 async dynamic address
 ip unnumbered ethernet 0

line 1
 autoselect ppp
 modem callin
 speed 19200
```

Remote Network Access Using PPP and Routing IP Example

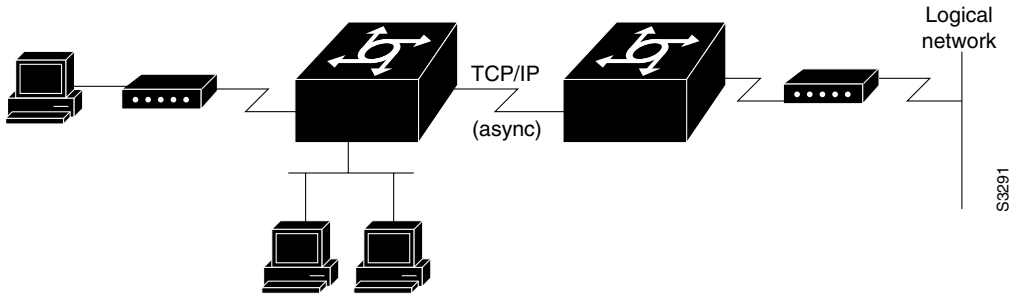
Figure 89 illustrates a network configuration that provides routing functionality, allowing routing updates to be passed across the asynchronous lines.

This network is composed of remote and local PCs connected via modem and network connections to an access server. This access server is connected to a second access server via an asynchronous line running TCP/IP. The second access server is connected to a local network via modem.

For this scenario, you will need to configure the following:

- An asynchronous line on both access servers configured to use PPP encapsulation
- An interface on both access servers for the modem connection and for this interface to be configured to accept incoming modem calls
- A default IP address for each incoming line
- IP routing on all configured interfaces

Figure 89 Routing on an Asynchronous Line Using PPP



The configuration is as follows:

```
interface async 1
 encapsulation ppp
 async mode interactive
 async default ip address 192.168.32.10
 async dynamic address
 ip unnumbered ethernet 0
 async dynamic routing
```

If you want to pass IP routing updates across the asynchronous link, enter the following commands:

```
line 1
 autoselect ppp
 modem callin
 speed 19200
```

Next, enter the following commands to configure the asynchronous lines between the access servers beginning in global configuration mode:

```
interface async 2
 async default ip address 192.168.32.55
 ip tcp header compression passive
```

Finally, configure routing as described in the *Cisco IOS IP Configuration Guide* using one of the following methods. The server can route packets three different ways.

- Use ARP, which is the default behavior.
- Use a default-gateway by entering the command **ip default-gateway** *x.x.x.x,* where *x.x.x.x* is the IP address of a locally attached router.
- Run an IP routing protocol such as Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Enhanced IGRP (EIGRP), or Open Shortest Path First (OSPF).

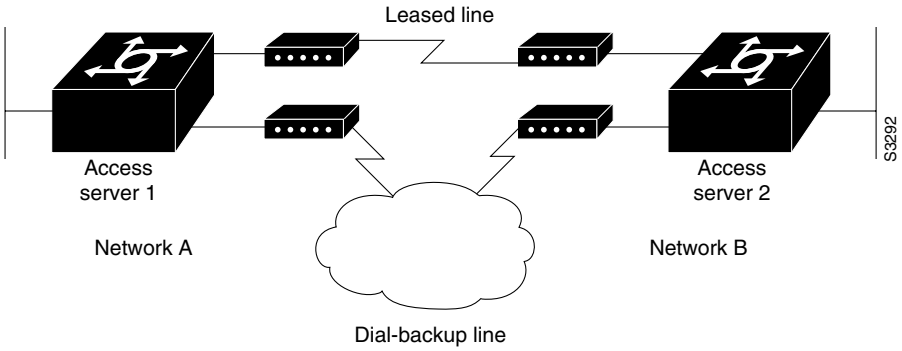
Remote Network Access Using a Leased Line with Dial-Backup and PPP Example

Figure 90 illustrates a scenario where two networks are connected via access servers on a leased line. Redundancy is provided by a dial-backup line over the PSTN so that if the primary leased line goes down, the dial-backup line will be automatically brought up to restore the connection. This configuration would be useful for using an auxiliary port as the backup port for a synchronous port.

For this scenario, you would need to configure the following:

- Two asynchronous interfaces on each access server
- Two modem interfaces
- A default IP address for each interface
- Dial-backup on one modem interface per access server
- An interface connecting to the related network of an access server

Figure 90 Asynchronous Leased Line with Backup



The configuration for this scenario follows:

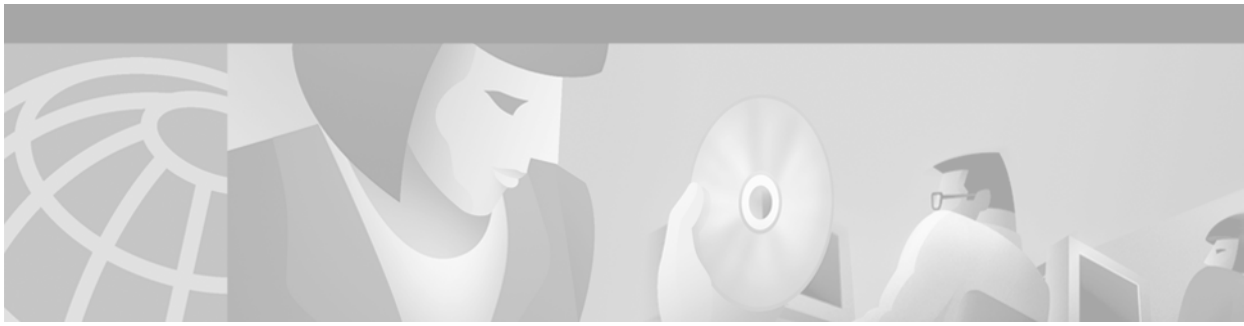
```
hostname routerA
!
username routerB password cisco
chat-script backup "" "AT" TIMEOUT 30 OK atdt\T TIMEOUT 30 CONNECT \c !
!
interface Serial0
 backup interface Async1
 ip address 192.168.222.12 255.255.255.0
!
interface Async1
 ip address 172.16.199.1 255.255.255.0
 encapsulation ppp
```

```
 async default ip address 172.16.199.2
 async dynamic address
 async dynamic routing
 async mode dedicated
 dialer in-band
 dialer map IP 172.16.199.2 name routerB modem-script backup broadcast 3241129
 dialer-group 1
 ppp authentication chap
!
 dialer-list 1 protocol ip permit
!
line aux 0
 modem InOut
 rxspeed 38400
 txspeed 38400
```

Multilink PPP Using Multiple Asynchronous Interfaces Example

The following example shows how to configure MLP using multiple asynchronous interfaces:

```
chat-script backup "" "AT" TIMEOUT 30 OK atdt\T TIMEOUT 30 CONNECT \c
!
ip address-pool local
ip pool foo 10.0.1.5 10.0.1.15
!
int as 1 (2, 3)
 no ip address
 dialer in-band
 encapsulation ppp
 ppp multilink
 dialer-rotary 1
!
interface dialer 1
 encaps ppp
 ip unnumbered ethernet 0
 peer default ip addr pool foo
 ppp authentication chap
 ppp multilink
 dialer in-band
 dialer map ip 10.200.100.9 name WAN-R3 modem-script backup broadcast 2322036
 dialer load-threshold 5 either
 dialer-group 1
!
dialer-list 1 protocol ip permit
!
line line 1 3
modem InOut
speed 115000
```



Configuring Media-Independent PPP and Multilink PPP

This chapter describes how to configure the PPP and Multilink PPP (MLP) features that can be configured on any interface. It includes the following main sections:

This chapter also describes address pooling for point-to-point links, which is available on all asynchronous serial, synchronous serial, and ISDN interfaces. See the chapter "Configuring Asynchronous SLIP and PPP" in this publication for information about PPP features and requirements that apply only to asynchronous lines and interfaces.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the PPP commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

PPP Encapsulation Overview

PPP, described in RFC 1661, encapsulates network layer protocol information over point-to-point links. You can configure PPP on the following types of physical interfaces:

- Asynchronous serial
- High-Speed Serial Interface (HSSI)
- ISDN
- Synchronous serial

By enabling PPP encapsulation on physical interfaces, PPP can also be in effect on calls placed by the dialer interfaces that use the physical interfaces.

The current implementation of PPP supports option 3, authentication using Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP), option 4, Link Quality Monitoring (LQM), and option 5, Magic Number configuration options. The software always sends option 5 and negotiates for options 3 and 4 if so configured. All other options are rejected.

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the **down-when-looped** command is configured, the router might shut down a link if it detects a loop.

The software provides the CHAP and PAP on serial interfaces running PPP encapsulation. For detailed information about authentication, refer to the *Cisco IOS Security Configuration Guide*.

Beginning with Cisco IOS Release 11.2 F, Cisco supported fast switching of incoming and outgoing DECnet and CLNS packets over PPP.

Configuring PPP and MLP

To configure PPP on a serial interface (including ISDN), perform the following task in interface configuration mode. This task is required for PPP encapsulation.

- Enabling PPP Encapsulation

You can also complete the tasks in the following sections; these tasks are optional but offer a variety of uses and enhancements for PPP on your systems and networks:

- Enabling CHAP or PAP Authentication
- Enabling Link Quality Monitoring
- Configuring Compression of PPP Data
- Configuring Microsoft Point-to-Point Compression
- Configuring IP Address Pooling
- Configuring PPP Reliable Link
- Disabling or Reenabling Peer Neighbor Routes
- Configuring PPP Half-Bridging
- Configuring Multilink PPP
- Configuring MLP Interleaving
- Enabling Distributed CEF Switching
- Creating a Multilink Bundle
- Assigning an Interface to a Multilink Bundle
- Disabling PPP Multilink Fragmentation
- Verifying the MLP Inverse Multiplexer Configuration

See the section "Monitoring and Maintaining PPP and MLP Interfaces" later in this chapter for tips on maintaining PPP. See the "Configuration Examples for PPP and MLP" at the end of this chapter for ideas on how to implement PPP and MLP in your network.

Enabling PPP Encapsulation

To enable PPP on serial lines to encapsulate IP and other network protocol datagrams, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.

PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. The **no keepalive command** can be used to disable echo requests.

Enabling CHAP or PAP Authentication

PPP with CHAP or PAP authentication is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP were originally specified in RFC 1334, and CHAP is updated in RFC 1994. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a *name*. This identification process prevents a router from placing another call to a router to which it is already connected, and also prevents unauthorized access.

Access control using CHAP or PAP is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

The required response has two parts:

- An encrypted version of the ID, a secret password, and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret password by performing the same encryption operation as indicated in the response and looking up the required host name or username. The secret passwords must be identical on the remote device and the local router.

Because this response is sent, the password is never sent in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only when a link is established. The local router or access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

To use CHAP or PAP, you must perform the following tasks:

- Enable PPP encapsulation.
- Enable CHAP or PAP on the interface.
- For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on an interface.

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp authentication {chap | chap pap | pap chap | pap} [if-needed] [list-name | default] [callin]`	Defines the authentication methods supported and the order in which they are used.

The **ppp authentication chap** optional keyword **if-needed** can be used only with Terminal Access Controller Access Control System (TACACS) or extended TACACS.

With authentication, authorization, and accounting (AAA) configured on the router and list names defined for AAA, the *list-name* optional keyword can be used with AAA/TACACS+.

Caution If you use a *list-name* that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

Add a **username** entry for each remote system from which the local router or access server requires authentication.

To specify the password to be used in CHAP or PAP caller identification, use the following command in global configuration mode:

Command	Purpose
`Router(config)# username name [user-maxlinks link-number] password secret`	Configures identification. Optionally, you can specify the maximum number of connections a user can establish. To use the **user-maxlinks** keyword, you must also use the **aaa authorization network default local** command and PPP encapsulation and name authentication on all the interfaces the user will be accessing.

Make sure this password does not include spaces or underscores.

To configure TACACS on a specific interface as an alternative to global host authentication, use one of the following commands in interface configuration mode:

Command	Purpose

`Router(config-if)# ppp use-tacacs [single-line]`
or
`Router(config-if)# aaa authentication ppp`
	Configures TACACS.

Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.

For an example of CHAP, see the section "CHAP with an Encrypted Password Examples" at the end of this chapter. CHAP is specified in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*.

Enabling Link Quality Monitoring

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the router will shut down the link. The percentages are calculated for both the incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent with the total number of packets and bytes received by the destination node. The incoming quality is calculated by comparing the total number of packets and bytes received with the total number of packets and bytes sent by the destination peer.

Note LQM is not compatible with Multilink PPP.

When LQM is enabled, Link Quality Reports (LQRs) are sent, in place of keepalives, every keepalive period. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

LQR is specified in RFC 1989, *PPP Link Quality Monitoring*.

To enable LQM on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp quality percentage`	Enables LQM on the interface.

The *percentage* argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and is taken down.

Configuring Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. PPP encapsulations support both predictor and Stacker compression algorithms.

If most of your traffic is already compressed files, do not use compression.

Most routers support software compression only, but in the Cisco 7000 series routers, hardware compression and distributed compression are also available, depending on the interface processor and compression service adapter hardware installed in the router.

To configure compression, complete the tasks in one of the following sections:

- Software Compression
- Hardware-Dependent Compression

Software Compression

Software compression is available in all router platforms. Software compression is performed by the main processor in the router.

Compression is performed in software and might significantly affect system performance. We recommend that you disable compression if the router CPU load exceeds 65 percent. To display the CPU load, use the **show process cpu** EXEC command.

To configure compression over PPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.
Step 2	`Router(config-if)# compress [predictor | stac | mppc [ignore-pfc]]`	Enables compression.

Hardware-Dependent Compression

When you configure Stacker compression on Cisco 7000 series routers with a 7000 Series Route Switch Processor (RSP7000), on Cisco 7200 series routers, and on Cisco 7500 series routers, there are three methods of compression: hardware compression, distributed compression, and software compression.

Hardware and distributed compression are available on routers that have the SA-Comp/1 and SA-Comp/4 data compression service adapters (CSAs). CSAs are available on Cisco 7200 series routers, on Cisco 7500 series routers with second-generation Versatile Interface Processors (VIP2s), and on Cisco 7000 series routers with the RSP7000 and 7000 Series Chassis Interface (RSP7000CI). (CSAs require VIP2 model VIP2-40.)

To configure hardware or distributed compression over PPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.

Step 2	**Cisco 7000 series with RSP7000 and Cisco 7500 series routers**
	`Router(config-if)# compress stac [distributed | software]`
	Cisco 7200 series routers
	`Router(config-if)# compress stac [csa slot | software]`
		Enables compression.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method:

- If the router contains a CSA, compression is performed in the CSA hardware (hardware compression).
- If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression).
- If the VIP2 is not available, compression is performed in the main processor of the router (software compression).

Using hardware compression in the CSA frees the main processor of the router for other tasks. You can also configure the router to use the VIP2 to perform compression by using the **distributed** option, or to use the main processor of the router by using the **software** option. If the VIP2 is not available, compression is performed in the main processor of the router.

When compression is performed in software installed in the main processor of the router, it might substantially affect system performance. We recommend that you disable compression in the main processor of the router if the router CPU load exceeds 40 percent. To display the CPU load, use the **show process cpu** EXEC command.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method.

Configuring Microsoft Point-to-Point Compression

Microsoft Point-to-Point Compression (MPPC) is a scheme used to compress PPP packets between Cisco and Microsoft client devices. The MPPC algorithm is designed to optimize bandwidth utilization in order to support multiple simultaneous connections. The MPPC algorithm uses a Lempel-Ziv (LZ)-based algorithm with a continuous history buffer called a dictionary.

The Compression Control Protocol (CCP) configuration option for MPPC is 18.

Exactly one MPPC datagram is encapsulated in the PPP information field. The PPP protocol field indicates the hexadecimal type of 00FD for all compressed datagrams. The maximum length of the MPPC datagram sent over PPP is the same as the MTU of the PPP interface; however, this length cannot be greater than 8192 bytes because the history buffer is limited to 8192 bytes. If compressing the data results in data expansion, the original data is sent as an uncompressed MPPC packet.

The history buffers between compressor and decompressor are synchronized by maintaining a 12-bit coherency count. If the decompressor detects that the coherency count is out of sequence, the following error recovery process is performed:

1. Reset Request (RR) packet is sent from the decompressor.
2. The compressor then flushes the history buffer and sets the flushed bit in the next packet it sends.
3. Upon receiving the flushed bit set packet, the decompressor flushes the history buffer.

Synchronization is achieved without CCP using the Reset Acknowledge (RA) packet, which can consume additional time.

Compression negotiation between a router and a Windows 95 client occurs through the following process:

1. Windows 95 sends a request for both STAC (option 17) and MPPC (option 18) compression.
2. The router sends a negative acknowledgment (NAK) requesting only MPPC.
3. Windows 95 resends the request for MPPC.
4. The router sends an acknowledgment (ACK) confirming MPPC compression negotiation.

MPPC Restrictions

The following restrictions apply to the MPPC feature:

- MPPC is supported only with PPP encapsulation.
- Compression can be processor intensive because it requires a reserved block of memory to maintain the history buffer. Do not enable modem or hardware compression because it may cause performance degradation, compression failure, or data expansion.
- Both ends of the point-to-point link must be using the same compression method (STAC, Predictor, or MPPC, for example).

Configuring MPPC

PPP encapsulation must be enabled before you can configure MPPC. For information on how to configure PPP encapsulation, see the section "Enabling PPP Encapsulation" earlier in this chapter.

There is only one command required to configure MPPC. The existing **compress** command supports the **mppc** keyword, which prepares the interface to initiate CCP and negotiates MPPC with the Microsoft client. To set MPPC once PPP encapsulation is configured on the router, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# compress [mppc [ignore-pfc]]`	Enables MPPC on the interface.

The **ignore-pfc** keyword instructs the router to ignore the protocol field compression flag negotiated by LCP. For example, the uncompressed standard protocol field value for IP is 0x0021 and 0x21 when compression is enabled. When the **ignore-pfc** option is enabled, the router will continue to use the uncompressed value (0x0021). Using the **ignore-pfc** option is helpful for some asynchronous driver devices that use an uncompressed protocol field (0x0021), even though the protocol field compression is negotiated between peers. displays protocol rejections when the **debug ppp negotiation** command is enabled. These errors can be remedied by setting the **ignore-pfc** option.

Sample debug ppp negotiation Command Output Showing Protocol Reject

```
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
```

Configuring IP Address Pooling

A point-to-point interface must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be obtained from a variety of sources. The address can be configured through the command line, entered with an EXEC-level command, provided by TACACS+ or the Dynamic Host Configuration Protocol (DHCP), or from a locally administered pool.

IP address pooling uses a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through IPCP address negotiation process. IP address pooling also enhances configuration flexibility by allowing multiple types of pooling to be active simultaneously.

See the chapter “Configuring Asynchronous SLIP and PPP” in this publication for additional information about address pooling on asynchronous interfaces and about the Serial Line Internet Protocol (SLIP).

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup—This method is used only if the peer requests an IP address, no other peer IP address has been assigned, and the interface is a member of a dialer group.
- PPP or SLIP EXEC command—An asynchronous dialup user can enter a peer IP address or host name when PPP or SLIP is invoked from the command line. The address is used for the current session and then discarded.
- IPCP negotiation—If the peer presents a peer IP address during IPCP address negotiation and no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Default IP address—The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.
- TACACS+ assigned IP address—During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that the user being authenticated on a dialup interface can use. This address overrides any default IP address and prevents pooling from taking place.
- DHCP retrieved IP address—If configured, the routers acts as a proxy client for the dialup user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.

- Local address pool—The local address pool contains a set of contiguous IP addresses (a maximum of 1024 addresses) stored in two queues. The free queue contains addresses available to be assigned and the used queue contains addresses that are in use. Addresses are stored to the free queue in first-in, first-out (FIFO) order to minimize the chance the address will be reused, and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.
- Chat script (asynchronous serial interfaces only)—The IP address in the **dialer map** command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- Virtual terminal/protocol translation—The translate command can define the peer IP address for a virtual terminal (pseudo asynchronous interface).
- The pool configured for the interface is used, unless TACACS+ returns a pool name as part of AAA. If no pool is associated with a given interface, the global pool named default is used.

Precedence Rules

The following precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address exists)
4. Address from an EXEC-level PPP or SLIP command, or from a chat script
5. Configured address from the **peer default ip address** command or address from the protocol **translate** command
6. Peer provided address from IPCP negotiation (not accepted unless no other address exists)

Interfaces Affected

Address pooling is available on all asynchronous serial, synchronous serial, ISDN BRI, and ISDN PRI interfaces that are running PPP.

Choosing the IP Address Assignment Method

The IP address pooling feature now allows configuration of a global default address pooling mechanism, per-interface configuration of the address pooling mechanism, and per-interface configuration of a specific address or pool name.

You can define the type of IP address pooling mechanism used on router interfaces in one or both of the ways described in the following sections:

- Defining the Global Default Address Pooling Mechanism
- Configuring IP Address Assignment

Defining the Global Default Address Pooling Mechanism

The global default mechanism applies to all point-to-point interfaces that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the global default mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform the tasks in one of following sections:

- Defining DHCP as the Global Default Mechanism
- Defining Local Address Pooling as the Global Default Mechanism

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface or you can configure the interface with a specific IP address to be used for dial-in peers.

You can also control the DHCP network discovery mechanism; see the following section for more information:

- Controlling DHCP Network Discovery

Defining DHCP as the Global Default Mechanism

DHCP specifies the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy-client—A Cisco access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. The DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

To enable DHCP as the global default mechanism, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip address-pool dhcp-proxy-client`	Specifies DHCP client-proxy as the global default mechanism.
Step 2	`Router(config)# ip dhcp-server [ip-address | name]`	(Optional) Specifies the IP address of a DHCP server for the proxy client to use.

In Step 2, you can provide as few as one or as many as ten DHCP servers for the proxy-client (the Cisco router or access server) to use. DHCP servers provide temporary IP addresses.

Defining Local Address Pooling as the Global Default Mechanism

To specify that the global default mechanism to use is local pooling, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip address-pool local`	Specifies local pooling as the global default mechanism.
Step 2	`Router(config)# ip local pool {named-address-pool | default} {first-IP-address [last-IP-address]} [group group-name] [cache-size size]}`	Creates one or more local IP address pools.

If no other pool is defined, a local pool called "default" is used. Optionally, you can associate an address pool with a named pool group.

Controlling DHCP Network Discovery

To allow peer routers to dynamically discover Domain Name System (DNS) and NetBIOS name server information configured on a DHCP server using PPP IP Control Protocol (IPCP) extensions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip dhcp-client network-discovery informs number-of-messages discovers number-of-messages period seconds`	Provides control of the DHCP network discovery mechanism by allowing the number of DHCP Inform and Discover messages to be sent, and a time-out period for retransmission, to be configured.

The **ip dhcp-client network-discovery** global configuration command provides a way to control the DHCP network discovery mechanism. The number of DHCP Inform or Discovery messages can be set to 1 or 2, which determines how many times the system sends the DHCP Inform or Discover messages before stopping network discovery. You can set a time-out period from 3 to 15 seconds, or leave the default time-out period at 15 seconds. Default for the **informs** and **discovers** keywords is 0, which disables the transmission of these messages.

Configuring IP Address Assignment

After you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can configure the few interfaces for which it is important to have a nondefault configuration. You can do any of the following;

- Define a nondefault address pool for use by a specific interface.
- Define DHCP on an interface even if you have defined local pooling as the global default mechanism.
- Specify one IP address to be assigned to all dial-in peers on an interface.
- Make temporary IP addresses available on a per-interface basis to asynchronous clients using SLIP or PPP.

To define a nondefault address pool for use on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip local pool {named-address-pool | default} {first-IP-address [last-IP-address]} [group group-name] [cache-size size]`	Creates one or more local IP address pools.
Step 2	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 3	`Router(config-if)# peer default ip address pool pool-name-list`	Specifies the pool or pools for the interface to use.

To define DHCP as the IP address mechanism for an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 2	`Router(config-if)# peer default ip address pool dhcp`	Specifies DHCP as the IP address mechanism on this interface.

To define a specific IP address to be assigned to all dial-in peers on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 2	`Router(config-if)# peer default ip address ip-address`	Specifies the IP address to assign.

Configuring PPP Reliable Link

PPP reliable link is Cisco's implementation of RFC 1663, *PPP Reliable Transmission*, which defines a method of negotiating and using Numbered Mode Link Access Procedure, Balanced (LAPB) to provide a reliable serial link. Numbered Mode LAPB provides retransmission of error packets across the serial link.

Although LAPB protocol overhead consumes some bandwidth, you can offset that consumption by the use of PPP compression over the reliable link. PPP compression is separately configurable and is not required for use of a reliable link.

Note PPP reliable link is available only on synchronous serial interfaces, including ISDN BRI and ISDN PRI interfaces. PPP reliable link cannot be used over V.120, and does not work with Multilink PPP.

To configure PPP reliable link on a specified interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp reliable-link`	Enables PPP reliable link.

Having reliable links enabled does not guarantee that all connections through the specified interface will in fact use reliable link. It only guarantees that the router will attempt to negotiate reliable link on this interface.

Troubleshooting PPP

You can troubleshoot PPP reliable link by using the **debug lapb** command and the **debug ppp negotiations**, **debug ppp errors**, and **debug ppp packets** commands. You can determine whether LAPB has been established on a connection by using the **show interface** command.

Disabling or Reenabling Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.

To disable this default behavior or to reenable it once it has been disabled, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no peer neighbor-route`	Disables creation of neighbor routes.
Step 2	`Router(config-if)# peer neighbor-route`	Reenables creation of neighbor routes.

Note If entered on a dialer or asynchronous group interface, this command affects all member interfaces.

Configuring PPP Half-Bridging

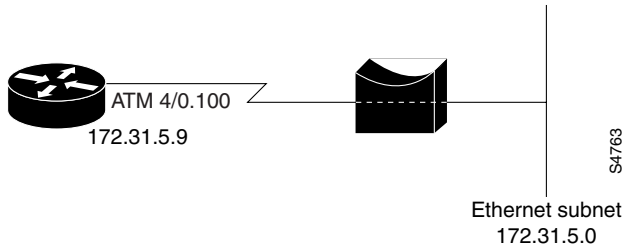
For situations in which a routed network needs connectivity to a remote bridged Ethernet network, a serial or ISDN interface can be configured to function as a PPP half-bridge. The line to the remote bridge functions as a virtual Ethernet interface, and the serial or ISDN interface on the router functions as a node on the same Ethernet subnetwork as the remote network.

The bridge sends bridge packets to the PPP half-bridge, which converts them to routed packets and forwards them to other router processes. Likewise, the PPP half-bridge converts routed packets to Ethernet bridge packets and sends them to the bridge on the same Ethernet subnetwork.

Note An interface cannot function as both a half-bridge and a bridge.

Figure 91 shows a router with a serial interface configured as a PPP half-bridge. The interface functions as a node on the Ethernet subnetwork with the bridge. Note that the serial interface has an IP address on the same Ethernet subnetwork as the bridge.

Figure 91 Router Serial Interface Configured as a Half-Bridge



Note The Cisco IOS software supports no more than one PPP half-bridge per Ethernet subnetwork.

To configure a serial interface to function as a half-bridge, use the following commands beginning in global configuration mode as appropriate for your network:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies the interface and begins interface configuration mode.

Step 2	`Router(config-if)# ppp bridge appletalk`
	`Router(config-if)# ppp bridge ip`
	`Router(config-if)# ppp bridge ipx [novell-ether | arpa | sap | snap]`
		Enables PPP half-bridging for one or more routed protocols: AppleTalk, IP, or Internet Protocol Exchange (IPX).

Step 3	`Router(config-if)# ip address n.n.n.n`
	`Router(config-if)# appletalk address network.node`
	`Router(config-if)# appletalk cable-range cable-range network.node`
	`Router(config-if)# ipx network network`
		Provides a protocol address on the same subnetwork as the remote network.

Note You must enter the **ppp bridge** command either when the interface is shut down or before you provide a protocol address for the interface.

For more information about AppleTalk addressing, refer to the "Configuring AppleTalk" chapter of the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*. For more information about IPX addresses and encapsulations, refer to the "Configuring Novell IPX" chapter of the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Configuring Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic. The Cisco implementation of MLP supports the fragmentation and packet sequencing specifications in RFC 1990. Additionally, you can change the default endpoint discriminator value that is supplied as part of user authentication. Refer to RFC 1990 for more information about the endpoint discriminator.

MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP is designed to work over synchronous and asynchronous serial and BRI and PRI types of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation.

Perform the tasks in the following sections, as required for your network, to configure MLP:

- Configuring MLP on Synchronous Interfaces
- Configuring MLP on Asynchronous Interfaces
- Configuring MLP on a Single ISDN BRI Interface
- Configuring MLP on Multiple ISDN BRI Interfaces
- Configuring MLP Using Multilink Group Interfaces
- Changing the Default Endpoint Discriminator

Configuring MLP on Synchronous Interfaces

To configure Multilink PPP on synchronous interfaces, you configure the synchronous interfaces to support PPP encapsulation and Multilink PPP.

To configure a synchronous interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies an asynchronous interface.
Step 2	`Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# no fair-queue`	Disables WFQ on the interface.
Step 5	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 6	`Router(config-if)# pulse-time seconds`	Enables pulsing DTR signal intervals on the interface.

Repeat these steps for additional synchronous interfaces, as needed.

Configuring MLP on Asynchronous Interfaces

To configure MLP on asynchronous interfaces, configure the asynchronous interfaces to support dial-on-demand routing (DDR) and PPP encapsulation, and then configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

To configure an asynchronous interface to support DDR and PPP encapsulation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface async number`	Specifies an asynchronous interface and begins interface configuration mode.
Step 2	`Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer in-band`	Enables DDR on the interface.
Step 5	`Router(config-if)# dialer rotary-group number`	Includes the interface in a specific dialer rotary group.

Repeat these steps for additional asynchronous interfaces, as needed.

At some point, adding more asynchronous interfaces does not improve performance, With the default maximum transmission unit (MTU) size, MLP should support three asynchronous interfaces using V.34 modems. However, packets might be dropped occasionally if the maximum transmission unit (MTU) size is small or large bursts of short frames occur.

To configure a dialer interface to support PPP encapsulation and Multilink PPP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Defines a dialer rotary group.
Step 2	`Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer in-band`	Enables DDR on the interface.
Step 5	`Router(config-if)# dialer load-threshold load [inbound | outbound | either]`	Configures bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.
Step 6	`Router(config-if)# ppp multilink`	Enables Multilink PPP.

Configuring MLP on a Single ISDN BRI Interface

To enable MLP on a single ISDN BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

To enable PPP on an ISDN BRI interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri number`	Specifies an interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Provides an appropriate protocol address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer idle-timeout seconds [inbound | either]`	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 5	`Router(config-if)# dialer load-threshold load`	Specifies the dialer load threshold for bringing up additional WAN links.
Step 6	`Router(config-if)# dialer map protocol next-hop-address [name hostname] [spc] [speed 56 | 64] [broadcast] [dial-string[:isdn-subaddress]]`	Configures the ISDN interface to call the remote site.
Step 7	`Router(config-if)# dialer-group group-number`	Controls access to this interface by adding it to a dialer access group.
Step 8	`Router(config-if)# ppp authentication pap`	(Optional) Enables PPP authentication.
Step 9	`Router(config-if)# ppp multilink`	Enables MLP on the dialer rotary group.

If you do not use PPP authentication procedures (Step 8), your telephone service must pass caller ID information.

The load threshold number is required. For an example of configuring MLP on a single ISDN BRI interface, see the section "MLP on One ISDN BRI Interface Example" at the end of this chapter.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer-load threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely, and the **dialer-load threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Configuring MLP on Multiple ISDN BRI Interfaces

To enable MLP on multiple ISDN BRI interfaces, set up a dialer rotary interface and configure it for Multilink PPP, and then configure the BRI interfaces separately and add them to the same rotary group.

To set up the dialer rotary interface for the BRI interfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Specifies the dialer rotary interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address address mask`	Specifies the protocol address for the dialer rotary interface.

	Command	Purpose
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer in-band`	Specifies in-band dialing.
Step 5	`Router(config-if)# dialer idle-timeout` *seconds* [**inbound** | **either**]	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, both inbound and outbound traffic will reset the dialer idle timer. Including the **inbound** keyword will cause only inbound traffic to reset the timer.
Step 6	`Router(config-if)# dialer map` *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]	Maps the next hop protocol address and name to the dial string needed to reach it.
Step 7	`Router(config-if)# dialer load-threshold` *load*	Specifies the dialer load threshold, using the same threshold as the individual BRI interfaces.
Step 8	`Router(config-if)# dialer-group` *number*	Controls access to this interface by adding it to a dialer access group.
Step 9	`Router(config-if)# ppp authentication chap`	(Optional) Enables PPP CHAP authentication.
Step 10	`Router(config-if)# ppp multilink`	Enables Multilink PPP.

If you do not use PPP authentication procedures (Step 10), your telephone service must pass caller ID information.

To configure each of the BRI interfaces to belong to the same rotary group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri` *number*	Specifies one of the BRI interfaces.
Step 2	`Router(config-if)# no ip address`	Specifies that it does not have an individual protocol address.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer idle-timeout` *seconds* [**inbound** | **either**]	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 5	`Router(config-if)# dialer rotary-group` *number*	Adds the interface to the rotary group.
Step 6	`Router(config-if)# dialer load-threshold` *load*	Specifies the dialer load threshold for bringing up additional WAN links.

Repeat Steps 1 through 6 for each BRI that you want to belong to the same dialer rotary group.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer load-threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely and the **dialer load-threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Note Previously, when MLP was used in a dialer profile, a virtual access interface was always created as the bundle. It was bound to both the B channel and the dialer profile interfaces after creation and cloning. The dialer profile interface could act as the bundle without help from a virtual access interface. But with the Dynamic Multiple Encapsulations feature available in Cisco IOS Release 12.1, it is no longer the virtual access interface that is added into the connected group of the dialer profile, but the dialer profile itself. The dialer profile becomes a connected member of its own connected group. See the "Dynamic Multiple Encapsulations over ISDN Example" in the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles" in this publication, for more information about dynamic multiple encapsulations and its relation to Multilink PPP.

For an example of configuring MLP on multiple ISDN BRI interfaces, see the section "MLP on Multiple ISDN BRI Interfaces Example" at the end of this chapter.

Configuring MLP Using Multilink Group Interfaces

MLP can be configured by assigning a multilink group to a virtual template configuration. Virtual templates allow a virtual access interface to dynamically clone interface parameters from the specified virtual template. If a multilink group is assigned to a virtual template, and then the virtual template is assigned to a physical interface, all links that pass through the physical interface will belong to the same multilink bundle.

A multilink group interface configuration will override a global multilink virtual template configured with the **multilink virtual template** command.

Multilink group interfaces can be used with ATM, PPP over Frame Relay, and serial interfaces.

To configure MLP using a multilink group interface, perform the following tasks:

- Configure the multilink group.
- Assign the multilink group to a virtual template.
- Configure the physical interface to use the virtual template.

To configure the multilink group, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# interface multilink group-number`	Creates a multilink bundle and enters multilink interface configuration mode to configure the bundle.
`Router(config-if)# ip address address mask`	Sets a primary IP address for an interface.
`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
`Router(config-if)# ppp multilink`	Enables MLP on an interface.

To assign the multilink group to a virtual template, perform the following task beginning in global configuration mode:

`Router(config)# interface virtual template number`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
`Router(config-if)# ppp multilink group group-number`	Restricts a physical link to joining only a designated multilink-group interface.

To configure the physical interface and assign the virtual template to it, perform the following task beginning in global configuration mode. This example is for an ATM interface. However, multilink group interfaces can also be used with PPP over Frame Relay interfaces and serial interfaces.

`Router(config)# interface atm` *interface-number.subinterface-number* `point-to-point`	Configures an ATM interface and enters interface configuration mode.
`Router(config-if)# pvc` *vpi/vci*	Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode.
`Router(config-if-atm-vc)# protocol ppp virtual-template` *name*	Configures VC multiplexed encapsulation on a PVC.

To see an example of how to configure MLP over an ATM PVC using a multilink group, see the section "MLP Using Multilink Group Interfaces over ATM Example" at the end of this chapter.

Changing the Default Endpoint Discriminator

By default, when the system negotiates use of MLP with the peer, the value that is supplied for the endpoint discriminator is the same as the username used for authentication. That username is configured for the interface by the Cisco IOS **ppp chap hostname** or **ppp pap sent-username** command, or defaults to the globally configured host name (or stack group name, if this interface is a Stack Group Bidding Protocol, or SGBP, group member).

To override or change the default endpoint discriminator, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink endpoint {hostname | ip` *IP-address* `| mac` *LAN-interface* `| none | phone` *telephone-number* `| string` *char-string*`}`	Overrides or changes the default endpoint discriminator the system uses when negotiating the use of MLP with the peer.

To see an example of how to change the default endpoint discriminator, see the section "Changing the Default Endpoint Discriminator Example" at the end of this chapter.

Configuring MLP Interleaving and Queueing

Interleaving on MLP allows large packets to be multilink encapsulated and fragmented into a small enough size to satisfy the delay requirements of real-time traffic; small real-time packets are not multilink encapsulated and are sent between fragments of the large packets. The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be sent earlier than other flows.

Weighted fair queueing on MLP works on the packet level, not at the level of multilink fragments. Thus, if a small real-time packet gets queued behind a larger best-effort packet and no special queue has been reserved for real-time packets, the small packet will be scheduled for transmission only after all the fragments of the larger packet are scheduled for transmission.

Weighted fair queueing is now supported on all interfaces that support Multilink PPP, including MLP virtual access interfaces and virtual interface templates. Weighted fair-queueing is enabled by default.

Fair queueing on MLP overcomes a prior restriction. Previously, fair queueing was not allowed on virtual access interfaces and virtual interface templates. Interleaving provides the delay bounds for delay-sensitive voice packets on a slow link that is used for other best-effort traffic.

Interleaving applies only to interfaces that can configure a multilink bundle interface. These restrictions include virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces.

Multilink and fair queueing are not supported when a multilink bundle is off-loaded to a different system using Multichassis Multilink PPP (MMP). Thus, interleaving is not supported in MMP networking designs.

MLP support for interleaving can be configured on virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces. To configure interleaving, complete the following tasks:

- Configure the dialer interface, BRI interface, PRI interface, or virtual template, as defined in the relevant chapters of this manual.
- Configure MLP and interleaving on the interface or template.

Note Fair queueing, which is enabled by default, must remain enabled on the interface.

Configuring MLP Interleaving

To configure MLP and interleaving on a configured and operational interface or virtual interface template, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 2	`Router(config-if)# ppp multilink interleave`	Enables interleaving of packets among the fragments of larger packets on an MLP bundle.
Step 3	`Router(config-if)# ppp multilink fragment delay` *milliseconds*	Specifies a maximum size, in units of time, for packet fragments on an MLP bundle.
Step 4	`Router(config-if)# ip rtp reserve` *lowest-udp-port range-of-ports* [*maximum-bandwidth*]	Reserves a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows.
Step 5	`Router(config-if)# exit`	Exits interface configuration mode.
Step 6	`Router(config)# multilink virtual-template 1`	For virtual templates only, applies the virtual template to the multilink bundle.[1]

1. This step is not used for ISDN or dialer interfaces.

Interleaving statistics can be displayed by using the **show interfaces** command, specifying the particular interface on which interleaving is enabled. Interleaving data is displayed only if there are interleaves. For example, the following line shows interleaves:

```
Output queue: 315/64/164974/31191 (size/threshold/drops/interleaves)
```

Configuring MLP Inverse Multiplexer and Distributed MLP

The distributed MLP feature combines T1/E1 lines in a VIP on a Cisco 7500 series router into a bundle that has the combined bandwidth of the multiple T1/E1 lines. This is done using a VIP MLP link. You choose the number of bundles and the number of T1/E1 lines in each bundle, which allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line without having to purchase a T3 line.

Nondistributed MLP can only perform limited links, with CPU usage quickly reaching 90% with only a few T1/E1 lines running MLP. With distributed MLP, you can increase the router's total capacity.

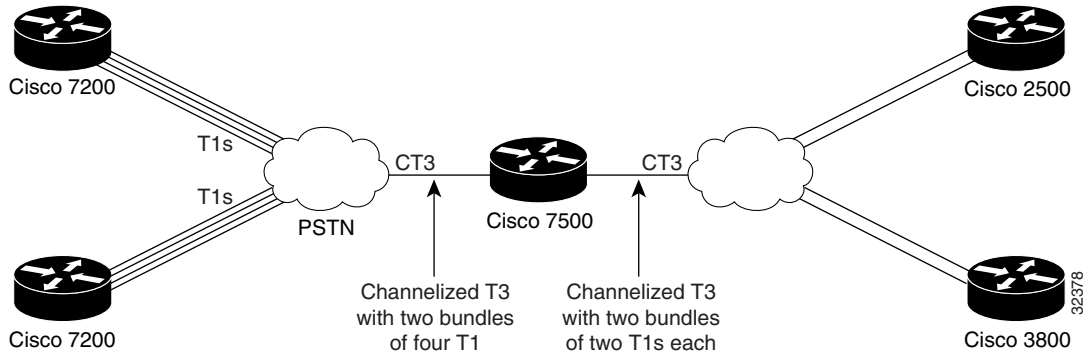
The MLP Inverse Multiplexer feature was designed for Internet service providers (ISPs) that want to have the bandwidth of multiple T1 lines with performance comparable to that of an inverse multiplexer without the need of buying standalone inverse-multiplexing equipment. A Cisco router supporting VIPs can bundle multiple T1 lines in a CT3 or CE3 interface. Bundling is more economical than purchasing an inverse multiplexer, and eliminates the need to configure another piece of equipment.

This feature supports the CT3 CE3 data rates without taxing the RSP and CPU by moving the data path to the VIP. This feature also allows remote sites to purchase multiple T1 lines instead of a T3 line, which is especially useful when the remote site does not need the bandwidth of an entire T3 line.

This feature allows multilink fragmentation to be disabled, so multilink packets are sent using Cisco Express Forwarding (CEF) on all platforms, if fragmentation is disabled. CEF is now supported with fragmentation enabled or disabled.

Figure 92 shows a typical network using a VIP MLP link. The Cisco 7500 series router is connected to the network with a CT3 line that has been configured with VIP MLP to carry two bundles of four T1 lines each. One of these bundles goes out to a Cisco 2500 series router and the other goes out to a Cisco 3800 series router.

Figure 92 Diagram of a Typical VIP MLP Topology



Before beginning the MLP Inverse Multiplexer configuration tasks, make note of the following prerequisites and restrictions.

Prerequisites

- Distributed CEF switching must be enabled for distributed MLP.
- One of the following port adapters is required:
  - CT3IP
  - PA-MC-T3

  - PA-MC-2T3+
  - PA-MC-E3
  - PA-MC-8T1
  - PA-MC-4T1
  - PA-MC-8E1
- All 16 E1s can be bundled from a PA-MC-E3 in a VIP4-80.

Restrictions

- The Multilink Inverse Multiplexer feature is supported only on the Cisco 7500 series routers.
- For bundles using IP, all lines in the bundle must have the same IP access list.
- Only one port adapter can be installed in a VIP.
- T1 and E1 lines cannot be mixed in a bundle.
- T1 lines in a bundle must have the same bandwidth.
- All lines in a bundle must have identical configurations.
- T1 lines can be combined in one bundle or up to 16 bundles per VIP.
- E1 lines can be combined in one bundle or up to 12 bundles per VIP.
- A maximum of eight T1 lines can be bundled on the VIP2-50 with two MB of SRAM.
- A maximum of 16 T1 lines can be bundled on the VIP2-50 with four or eight MB of SRAM.
- A maximum of 12 E1 lines can be bundled on the VIP2-50 with four or eight MB of SRAM.
- A maximum of 40 T1 lines can be bundled on the VIP4-80.
- Hardware compression is not supported.
- Encryption is not supported.
- Fancy/custom queueing is supported.
- MLP fragmentation is supported.
- Software compression is not recommended because CPU usage would negate performance gains.
- The maximum differential delay supported is 50 milliseconds.
- VIP CEF is limited to IP only; all other protocols are sent to the RSP.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation may result in better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation may be outweighed by the added load on the CPU.

To configure a multilink bundle, perform the tasks in the following sections:

- Enabling Distributed CEF Switching (Required for Distributed MLP)
- Creating a Multilink Bundle (Required)
- Assigning an Interface to a Multilink Bundle (Required)
- Disabling PPP Multilink Fragmentation (Optional)
- Verifying the MLP Inverse Multiplexer Configuration (Optional)

Enabling Distributed CEF Switching

To enable distributed MLP, first enable distributed CEF (dCEF) switching using the following command in global configuration mode:

Command	Purpose
`Router(config)# ip cef distributed`	Enables dCEF switching.

Creating a Multilink Bundle

To create a multilink bundle, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink` *group-number*	Assigns a multilink group number and begins interface configuration mode.
Step 2	`Router(config-if)# ip address` *address mask*	Assigns an IP address to the multilink interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp multilink`	Enables Multilink PPP.

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no ip address`	Removes any specified IP address.
Step 2	`Router(config-if)# keepalive`	Sets the frequency of keepalive packets.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp multilink group` *group-number*	Restricts a physical link to joining only the designated multilink-group interface.
Step 5	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 6	`Router(config-if)# ppp authentication chap`	(Optional) Enables CHAP authentication.
Step 7	`Router(config-if)# pulse-time` *seconds*	(Optional) Configures DTR signal pulsing.

Disabling PPP Multilink Fragmentation

By default, PPP multilink fragmentation is enabled. To disable PPP multilink fragmentation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink fragment disable`	(Optional) Disables PPP multilink fragmentation.

Verifying the MLP Inverse Multiplexer Configuration

To display information about the newly created multilink bundle, use the **show ppp multilink** command in EXEC mode:

```
Router# show ppp multilink

Multilink1, bundle name is group1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links:4 active, 0 inactive (max not set, min not set)
 Serial1/0/0:1
 Serial1/0/0/:2
 Serial1/0/0/:3
 Serial1/0/0/:4
```

Monitoring and Maintaining PPP and MLP Interfaces

To monitor and maintain virtual interfaces, use the following command in EXEC mode:

Command	Purpose
`Router> show ppp multilink`	Displays MLP and MMP bundle information.

Configuration Examples for PPP and MLP

The following sections provide various PPP configuration examples:

- CHAP with an Encrypted Password Examples
- User Maximum Links Configuration Example
- MPPC Interface Configuration Examples
- IP Address Pooling Example
- DHCP Network Control Example
- PPP Reliable Link Examples
- MLP Examples
- MLP Interleaving and Queueing for Real-Time Traffic Example

- T3 Controller Configuration for an MLP Multilink Inverse Multiplexer Example
- Multilink Interface Configuration for Distributed MLP Example

CHAP with an Encrypted Password Examples

The following examples show how to enable CHAP on serial interface 0 of three devices:

Configuration of Router yyy

```
hostname yyy
interface serial 0
 encapsulation ppp
 ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

Configuration of Router xxx

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password secretxy
username zzz password secretxz
```

Configuration of Router zzz

```
hostname zzz
interface serial 0
 encapsulation ppp
 ppp authentication chap
username xxx password secretxz
username yyy password secretzy
```

When you look at the configuration file, the passwords will be encrypted and the display will look similar to the following:

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

User Maximum Links Configuration Example

The following example shows how to configure the username sTephen and establish a maximum of five connections. sTephen can connect through serial interface 1/0, which has a dialer map configured for it, or through PRI interface 0/0:23, which has dialer profile interface 0 dedicated to it.

The **aaa authorization network default local** command must be configured. PPP encapsulation and authentication must be enabled on all the interfaces that sTephen can connect to.

```
aaa new-model
aaa authorization network default local
enable secret saintstephen
enable password witharose
!
username sTephen user-maxlinks 5 password gardenhegoes
```

```
!
interface Serial0/0:23
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap
 ppp multilink
!
interface Serial1/0
 ip address 10.2.2.4 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer map ip 10.2.2.13 name sTephen 12345
 dialer-group 1
 ppp authentication chap
!
interface Dialer0
 ip address 10.1.1.4 255.255.255.0
 encapsulation ppp
 dialer remote-name sTephen
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp authentication chap
 ppp multilink
!
dialer-list 1 protocol ip permit
```

MPPC Interface Configuration Examples

The following example configures asynchronous interface 1 to implement MPPC and ignore the protocol field compression flag negotiated by LCP:

```
interface async1
 ip unnumbered ethernet0
 encapsulation ppp
 async default routing
 async dynamic routing
 async mode interactive
 peer default ip address 172.21.71.74
 compress mppc ignore-pfc
```

The following example creates a virtual access interface (virtual-template interface 1) and serial interface 0, which is configured for X.25 encapsulation. MPPC values are configured on the virtual-template interface and will ignore the negotiated protocol field compression flag.

```
interface ethernet0
 ip address 172.20.30.102 255.255.255.0
!
interface virtual-template1
 ip unnumbered ethernet0
 peer default ip address pool vtemp1
 compress mppc ignore-pfc
!
interface serial0
 no ipaddress
no ip mroute-cache
encapsulation x25
x25 win 7
x25 winout 7
x25 ips 512
x25 ops 512
```

```
clock rate 50000
!
ip local pool vtemp1 172.20.30.103 172.20.30.104
ip route 0.0.0.0 0.0.0.0 172.20.30.1
!
translate x25 31320000000000 virtual-template 1
```

IP Address Pooling Example

The following example configures a modem to dial in to a Cisco access server and obtain an IP address from the DHCP server. This configuration allows the user to log in and browse an NT network. Notice that the dialer 1 and group-async 1 interfaces are configured with the **ip unnumbered loopback** command, so that the broadcast can find the dialup clients and the client can see the NT network.

```
!
hostname secret
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default if-needed local
aaa authentication ppp chap local
enable secret 5 encrypted-secret
enable password EPassWd1
!
username User1 password 0 PassWd2
username User2 password 0 PassWd3
username User3 password 0 PassWd4
no ip domain-lookup
ip dhcp-server 10.47.0.131
async-bootp gateway 10.47.0.1
async-bootp nbns-server 10.47.0.131
isdn switch-type primary-4ess
!
!
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
 linecode b8zs
!
interface loopback 0
 ip address 10.47.252.254 255.255.252.0
!
interface ethernet 0
 ip address 10.47.0.5 255.255.252.0
 ip helper-address 10.47.0.131
 ip helper-address 10.47.0.255
 no ip route-cache
 no ip mroute-cache
!
interface serial 0
 no ip address
 no ip mroute-cache
 shutdown
!
```

```
interface serial 1
 no ip address
 shutdown
!
interface serial 0:23
 no ip address
 encapsulation ppp
 no ip mroute-cache
 dialer rotary-group 1
 dialer-group 1
 isdn incoming-voice modem
 no fair-queue
 no cdp enable
!
interface group-async 1
 ip unnumbered loopback 0
 ip helper-address 10.47.0.131
 ip tcp header-compression passive
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 async mode interactive
 peer default ip address dhcp
 no fair-queue
 no cdp enable
 ppp authentication chap
 group-range 1 24
!
interface dialer 1
 ip unnumbered loopback 0
 encapsulation ppp
 dialer in-band
 dialer-group 1
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink
!
router ospf 172
 redistribute connected subnets
 redistribute static
 network 10.47.0.0 0.0.3.255 area 0
 network 10.47.156.0 0.0.3.255 area 0
 network 10.47.168.0 0.0.3.255 area 0
 network 10.47.252.0 0.0.3.255 area 0
!
ip local pool RemotePool 10.47.252.1 10.47.252.24
ip classless
ip route 10.0.140.0 255.255.255.0 10.59.254.254
ip route 10.2.140.0 255.255.255.0 10.59.254.254
ip route 10.40.0.0 255.255.0.0 10.59.254.254
ip route 10.59.254.0 255.255.255.0 10.59.254.254
ip route 172.23.0.0 255.255.0.0 10.59.254.254
ip route 192.168.0.0 255.255.0.0 10.59.254.254
ip ospf name-lookup
no logging buffered
access-list 101 deny ip any host 255.255.255.255
access-list 101 deny ospf any any
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
snmp-server community public RO
!
```

```
line con 0
line 1 24
 autoselect during-login
 autoselect ppp
 modem InOut
 transport input all
line aux 0
line vty 0 4
 password PassWd5
!
scheduler interval 100
end
```

DHCP Network Control Example

The following partial example adds the **ip dhcp-client network-discovery** command to the previous "IP Address Pooling Example" to allow peer routers to more dynamically discover DNS and NetBIOS name servers. If the **ip dhcp-client network-discovery** command is disabled, the system falls back to the static configurations made using the **async-bootp dns-server** and **async-bootp nb-server** global configuration commands.

```
!
hostname secret
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default if-needed local
aaa authentication ppp chap local
enable secret 5 encrypted-secret
enable password EPassWd1
!
username User1 password 0 PassWd2
username User2 password 0 PassWd3
username User3 password 0 PassWd4
no ip domain-lookup
ip dhcp-server 10.47.0.131
ip dhcp-client network-discovery informs 2 discovers 2 period 12
async-bootp gateway 10.47.0.1
async-bootp nbns-server 10.47.0.131
isdn switch-type primary-4ess
.
.
.
```

PPP Reliable Link Examples

The following example enables PPP reliable link and STAC compression on BRI 0:

```
interface BRI0
 description Enables stac compression on BRI 0
 ip address 172.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 172.1.1.2 name baseball 14195386368
 compress stac
 ppp authentication chap
 dialer-group 1
 ppp reliable-link
```

The following example shows output of the **show interfaces** command when PPP reliable link is enabled. The LAPB output lines indicate that PPP reliable link is provided over LAPB.

```
Router# show interfaces serial 0

Serial0 is up, line protocol is up
  Hardware is HD64570
  Description: connects to enkidu s 0
  Internet address is 172.21.10.10/8
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  LCP Open
  Open: IPCP, CDP
  LAPB DTE, state CONNECT, modulo 8, k 7, N1 12048, N2 20
      T1 3000, T2 0, interface outage (partial T3) 0, T4 0, PPP over LAPB
      VS 1, VR 1, tx NR 1, Remote VR 1, Retransmissions 0
      Queues: U/S frames 0, I frames 0, unack. 0, reTx 0
      IFRAMEs 1017/1017 RNRs 0/0 REJs 0/0 SABM/Es 1/1 FRMRs 0/0 DISCs 0/0
  Last input 00:00:18, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 (size/threshold/drops)
     Conversations  0/1 (active/max active)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 3000 bits/sec, 4 packets/sec
  5 minute output rate 3000 bits/sec, 7 packets/sec
     1365 packets input, 107665 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
     2064 packets output, 109207 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

MLP Examples

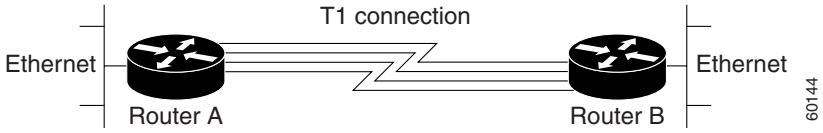
This section contains the following MLP examples:

- MLP on Synchronous Serial Interfaces Example
- MLP on One ISDN BRI Interface Example
- MLP on Multiple ISDN BRI Interfaces Example
- MLP Using Multilink Group Interfaces over ATM Example
- Changing the Default Endpoint Discriminator Example

MLP on Synchronous Serial Interfaces Example

MLP provides characteristics most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure 93 shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

Figure 93 Inverse Multiplexing Application Using Multilink PPP



The following example shows the configuration commands used to create the inverse multiplexing application:

Router A Configuration

```
hostname RouterA
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.1.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Router B Configuration

```
hostname RouterB
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.2.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

MLP on One ISDN BRI Interface Example

The following example enables MLP on BRI interface 0. Because an ISDN interface is a rotary group by default, when one BRI is configured, no dialer rotary group configuration is required.

```
interface bri 0
 description connected to ntt 81012345678902
 ip address 172.31.1.7 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 30
 dialer load-threshold 40 either
 dialer map ip 172.31.1.8 name atlanta 81012345678901
```

```
 dialer-group 1
 ppp authentication pap
 ppp multilink
```

MLP on Multiple ISDN BRI Interfaces Example

The following example configures multiple ISDN BRI interfaces to belong to the same dialer rotary group for Multilink PPP. The **dialer rotary-group** command is used to assign each of the ISDN BRI interfaces to that dialer rotary group.

```
interface BRI0
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI1
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI2
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface Dialer0
 ip address 10.0.0.2 255.0.0.0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 500
 dialer map ip 10.0.0.1 name atlanta broadcast 81012345678901
 dialer load-threshold 30 either
 dialer-group 1
 ppp authentication chap
 ppp multilink
```

MLP Using Multilink Group Interfaces over ATM Example

The following example configures MLP over an ATM PVC using a multilink group:

```
interface multilink 1
 ip address 10.200.83.106 255.255.255.252
 ip tcp header-compression iphc-format delay 20000
 service policy output xyz
 encapsulation ppp
 ppp multilink
 ppp multilink fragment delay 10
 ppp multilink interleave
 ppp timeout multilink link remove 10
 ip rtp header-compression iphc-format

interface virtual-template 3
 bandwidth 128
 ppp multilink group 1
```

```
interface atm 4/0.1 point-to-point
 pvc 0/32
 abr 100 80
 protocol ppp virtual-template 3
```

Changing the Default Endpoint Discriminator Example

The following partial example changes the MLP endpoint discriminator from the default CHAP host name C-host1 to the E.164-compliant telephone number `1 603 555-1212`:

```
.
.
.
interface dialer 0
 ip address 10.1.1.4 255.255.255.0
 encapsulation ppp
 dialer remote-name R-host1
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp chap hostname C-host1
 ppp multilink endpoint phone 16035551212
.
.
.
```

MLP Interleaving and Queueing for Real-Time Traffic Example

The following example defines a virtual interface template that enables MLP interleaving and a maximum real-time traffic delay of 20 milliseconds, and then applies that virtual template to the MLP bundle:

```
interface virtual-template 1
 ip unnumbered ethernet 0
 ppp multilink
 ppp multilink interleave
 ppp multilink fragment delay 20
 ip rtp interleave 32768 20 1000
multilink virtual-template 1
```

The following example enables MLP interleaving on a dialer interface that controls a rotary group of BRI interfaces. This configuration permits IP packets to trigger calls.

```
interface BRI 0
 description connected into a rotary group
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 2
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 3
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
```

```
interface BRI 4
 encapsulation ppp
 dialer rotary-group 1
!
interface Dialer 0
 description Dialer group controlling the BRIs
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name angus 14802616900
 dialer-group 1
 ppp authentication chap
! Enables Multilink PPP interleaving on the dialer interface and reserves
! a special queue.
 ppp multilink
 ppp multilink interleave
 ip rtp reserve 32768 20 1000
! Keeps fragments of large packets small enough to ensure delay of 20 ms or less.
 ppp multilink fragment delay 20
dialer-list 1 protocol ip permit
```

T3 Controller Configuration for an MLP Multilink Inverse Multiplexer Example

In the following example, the T3 controller is configured and four channelized interfaces are created:

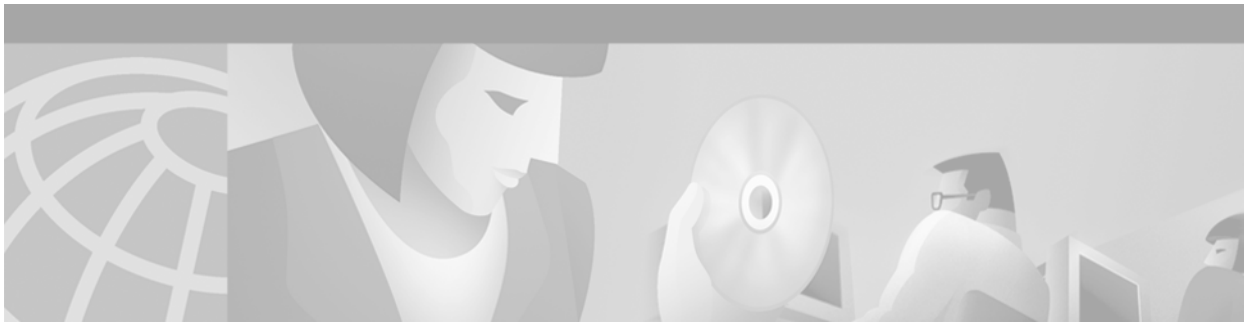
```
controller T3 1/0/0
framing m23
cablelength 10
t1 1 timeslots 1-24
t1 2 timeslots 1-24
t1 3 timeslots 1-24
t1 4 timeslots 1-24
```

Multilink Interface Configuration for Distributed MLP Example

In the following example, four multilink interfaces are created with distributed CEF switching and MLP enabled. Each of the newly created interfaces is added to a multilink bundle.

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hosstname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:1
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:2
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
```

```
interface serial 1/0/0/:3
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:4
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1
```



Configuring Multichassis Multilink PPP

This chapter describes how to configure Multichassis Multilink PPP (MLP). It includes the following main sections:

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the MMP commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Multichassis Multilink PPP Overview

Prior to Release 11.2, Cisco IOS supported Multilink PPP (MLP). Beginning with Release 11.2, Cisco IOS software also supports Multichassis Multilink PPP (MMP).

MLP provides the capability of splitting and recombining packets to a single end system across a logical pipe (also called a *bundle*) formed by multiple links. MMP provides bandwidth on demand and reduces transmission latency across WAN links.

MMP, however, provides the additional capability for links to terminate at multiple routers with different remote addresses. MMP can also handle both analog and digital traffic.

MLP is intended for situations with large pools of dial-in users, in which a single chassis cannot provide enough dial ports. This feature allows companies to provide a single dialup number to its users and to apply the same solution to analog and digital calls. This feature allows Internet service providers (ISPs), for example, to allocate a single ISDN rotary number to several ISDN PRIs across several routers. This capability allows for easy expansion and scalability and for assured fault tolerance and redundancy.

MMP allows network access servers to be stacked together and to appear as a single network access server chassis so that if one network access server fails, another network access server in the stack can accept calls.

With large-scale dial-out, these features are available for both outgoing and incoming calls.

Stack Groups

Routers or access servers are configured to belong to groups of peers called *stack groups*. All members of the stack group are peers; stack groups do not need a permanent lead router. Any stack group member can answer calls coming from a single access number, which is usually an ISDN PRI hunt group. Calls can come in from remote user devices, such as routers, modems, ISDN terminal adapters, and PC cards.

Once a connection is established with one member of a stack group, that member owns the call. If a second call comes in from the same client and a different router answers the call, the router establishes a tunnel and forwards all packets that belong to the call to the router that owns the call. Establishing a tunnel and forwarding calls through it to the router that owns the call is sometimes called *projecting the PPP link to the call master.*

If a more powerful router is available, it can be configured as a member of the stack group and the other stack group members can establish tunnels and forward all calls to it. In such a case, the other stack group members are just answering calls and forwarding traffic to the more powerful *offload* router.

Note High-latency WAN lines between stack group members can make stack group operation inefficient.

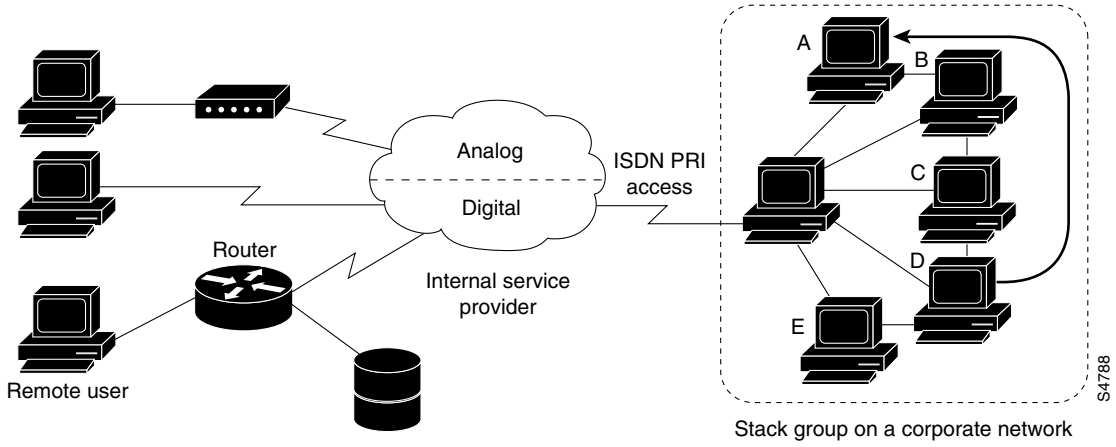
Call Handling and Bidding

MMP call handling, bidding, and Layer 2 forwarding operations in the stack group proceed as follows:

1. When the first call comes in to the stack group, router A answers.
2. In the bidding, router A wins because it already has the call. Router A becomes the *call-master* for that session with the remote device. (Router A might also be called the *host to the master bundle interface*.)
3. When the remote device that initiated the call needs more bandwidth, it makes a second MLP call to the group.
4. When the second call comes in, router D answers it and informs the stack group. Router A wins the bidding because it already is handling the session with that remote device.
5. Router D establishes a tunnel to router A and forwards the raw PPP data to router A.
6. Router A reassembles and resequences the packets.
7. If more calls come in to router D and they too belong to router A, the tunnel between routers A and D enlarges to handle the added traffic. Router D does not establish an additional tunnel to router A.
8. If more calls come in and are answered by any other router, that router also establishes a tunnel to router A and forwards the raw PPP data.
9. The reassembled data is passed on the corporate network as if it had all come through one physical link.

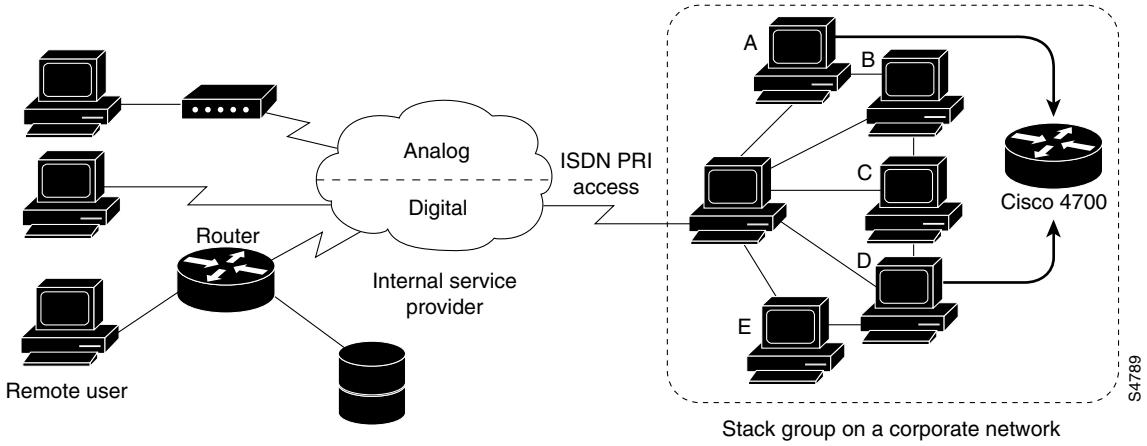
Figure 94 shows the call handling an bidding process in a typical MLP scenario.

Figure 94 Typical MLP Scenario



In contrast to Figure 94, Figure 95 features an offload router. Access servers that belong to a stack group answer calls, establish tunnels, and forward calls to a Cisco 4700 router that wins the bidding and is the call master for all the calls. The Cisco 4700 reassembles and resequences all the packets that come in through the stack group.

Figure 95 MLP with an Offload Router as a Stack Group Member



Note You can build stack groups using different access-server, switching, and router platforms. However, universal access servers such as the Cisco AS5200 should not be combined with ISDN-only access servers such as the Cisco 4000 series platform. Because calls from the central office are allocated in an arbitrary way, this combination could result in an analog call being delivered to a digital-only access server, which would not be able to handle the call.

MMP support on a group of routers requires that each router be configured to support the following:

- Multilink PPP
- Stack Group Bidding Protocol (SGBP)
- Virtual template used for cloning interface configuration to support MMP

MMP is supported on the Cisco 2500, 4500, and 7500 series platforms and on synchronous serial, asynchronous serial, ISDN BRI, ISDN PRI, and dialer interfaces.

MMP does not require reconfiguration of telephone company switches.

Dialer profiles are not supported for SGBP (Stack Group Bidding Protocol).

How to Configure MMP

To configure MMP, perform the tasks in the following sections, in the order listed:

- Configuring the Stack Group and Identifying Members (Required)
- Configuring a Virtual Template and Creating a Virtual Template Interface (Required)

See the section "Monitoring and Maintaining MMP Virtual Interfaces" later in this chapter for tips on maintaining MMP. See the examples in the section "Configuration Examples for MMP" later in this chapter for ideas on how to configure MMP in your network.

Configuring the Stack Group and Identifying Members

To configure the stack group on the router, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username name password password`	Creates authentication credentials for the stack group.
Step 2	`Router(config)# sgbp group name`	Creates the stack group and assign this router to it.
Step 3	`Router(config)# sgbp member peer-name [peer-ip-address]`	Specifies a peer member of the stack group.

Repeat these steps for each additional stack group peer.

Note Only one stack group can be configured per access server or router.

Configuring a Virtual Template and Creating a Virtual Template Interface

You need to configure a virtual template for MMP when asynchronous or synchronous serial interfaces are used, but dialers are not defined. When dialers are configured on the physical interfaces, do not specify a virtual template interface.

To configure a virtual template for any nondialer interfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# multilink virtual-template` *number*	Defines a virtual template for the stack group. This step is not required if ISDN interfaces or other dialers are configured and used by the physical interfaces.
Step 2	`Router(config)# ip local pool default` *ip-address*	Specifies an IP address pool by using any pooling mechanism—for example, IP local pooling or Dynamic Host Configuration Protocol (DHCP) pooling.
Step 3	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode. This step is not required if ISDN interfaces or other dialers are configured and used by the physical interfaces.
Step 4	`Router(config-if)# ip unnumbered ethernet 0`	Specifies unnumbered IP.
Step 5	`Router(config-if)# no ip route-cache`	Disables fast switching, which enables per-packet load sharing and enhances performance on slower serial links.
Step 6	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template interface.
Step 7	`Router(config-if)# ppp multilink`	Enables Multilink PPP on the virtual template interface.
Step 8	`Router(config-if)# ppp authentication chap`	Enables PPP authentication on the virtual template interface.

If dialers are or will be configured on the physical interfaces, the **ip unnumbered** command, mentioned in Step 4, will be used in configuring the dialer interface. For examples that show MMP configured with and without dialers, see the "Configuration Examples for MMP" at the end of this chapter.

Note Never define a specific IP address on the virtual template because projected virtual access interfaces are always cloned from the virtual template interface. If a subsequent PPP link also gets projected to a stack member with a virtual access interface already cloned and active, we will have identical IP addresses will be on the two virtual interfaces. IP will erroneously route between them.

For more information about address pooling, see the "Configuring Media-Independent PPP and Multilink PPP" chapter.

Monitoring and Maintaining MMP Virtual Interfaces

To monitor and maintain virtual interfaces, use any of the following commands in EXEC mode:

Command	Purpose
`Router> show ppp multilink`	Displays MLP and MMP bundle information.
`Router> show sgbp`	Displays the status of the stack group members.
`Router> show sgbp queries`	Displays the current seed bid value.

Configuration Examples for MMP

The following sections provide w MMP configuration examples without and with dialers:

- MMP Using PRI But No Dialers
- MMP with Dialers
- MMP with Offload Server

MMP Using PRI But No Dialers

The following example shows the configuration of MMP when no dialers are involved. Comments in the configuration discuss the commands. Variations are shown for a Cisco AS5200 access server or Cisco 4000 series router and for an E1 or T1 controller.

```
sgbp group stackq
sgbp member systemb 10.1.1.2
sgbp member systemc 10.1.1.3

username stackq password therock
! First make sure the multilink virtual template number is defined globally on
¡ each router that is a member of the stack group.
multilink virtual-template 1

! If you have not configured any dialer interfaces for the physical interfaces in
! question (PRI, BRI, async, sync serial), you can define a virtual template.

interface virtual-template 1
 ip unnumbered e0
 no ip route-cache
ppp authentication chap
 ppp multilink

! Never define a specific IP address on the virtual template because projected
! virtual access interfaces are always cloned from the virtual template interface.
! If a subsequent PPP link also gets projected to a stack member with a virtual
! access interface already cloned and active, identical IP addresses will be on
! on the two virtual interfaces. IP will erroneously route between them.

! On an AS5200 or 4XXX platform.
! On a TI controller.
!
controller T1 0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24
!
interface serial 0:23
 no ip address
 encapsulation ppp
 no ip route-cache
 ppp authentication chap
 ppp multilink
!
! On an E1 controller.
!
controller E1 0
 framing crc4
 linecode hdb3
 pri-group timeslots 1-31
```

```
interface serial 0:15
 no ip address
 encapsulation ppp
 no ip route-cache
 ppp authentication chap
 ppp multilink
```

MMP with Dialers

When dialers are configured on the physical interfaces and when the interface itself is a dialer, do not specify a virtual template interface. For dialers, you only need to define the stack group name, common password, and its members across all the stack members. No virtual template interface is defined at all.

Only the PPP commands in dialer interface configuration are applied to the bundle interface. Subsequent projected PPP links are also cloned with the PPP commands from the dialer interface.

Dialer profiles are not supported for SGBP (Stack Group Bidding Protocol).

This section includes the following examples:

- MMP with Explicitly Defined Dialer
- MMP with ISDN PRI but No Explicitly Defined Dialer

MMP with Explicitly Defined Dialer

The following example includes a dialer that is explicitly specified by the **interface dialer** command and configured by the commands that immediately follow:

```
sgbp group stackq
sgbp member systemb 10.1.1.2
sgbp member systemc 10.1.1.3

username stackq password therock

interface dialer 1
 ip unnumbered e0
 dialer map .....
 encapsulation ppp
 ppp authentication chap
 dialer-group 1
 ppp multilink
!
! On a T1 controller
controller T1 0
 framing esf
 linecode b8zs
 pri-group timeslots 1-24

interface Serial0:23
 no ip address
 encapsulation ppp
 dialer in-band
 dialer rotary 1
 dialer-group 1
!
! Or on an E1 Controller
controller E1 0
 framing crc4
 linecode hdb3
```

```
 pri-group timeslots 1-31

interface serial 0:15
 no ip address
 encapsulation ppp
 no ip route-cache
 ppp authentication chap
 ppp multilink
```

MMP with ISDN PRI but No Explicitly Defined Dialer

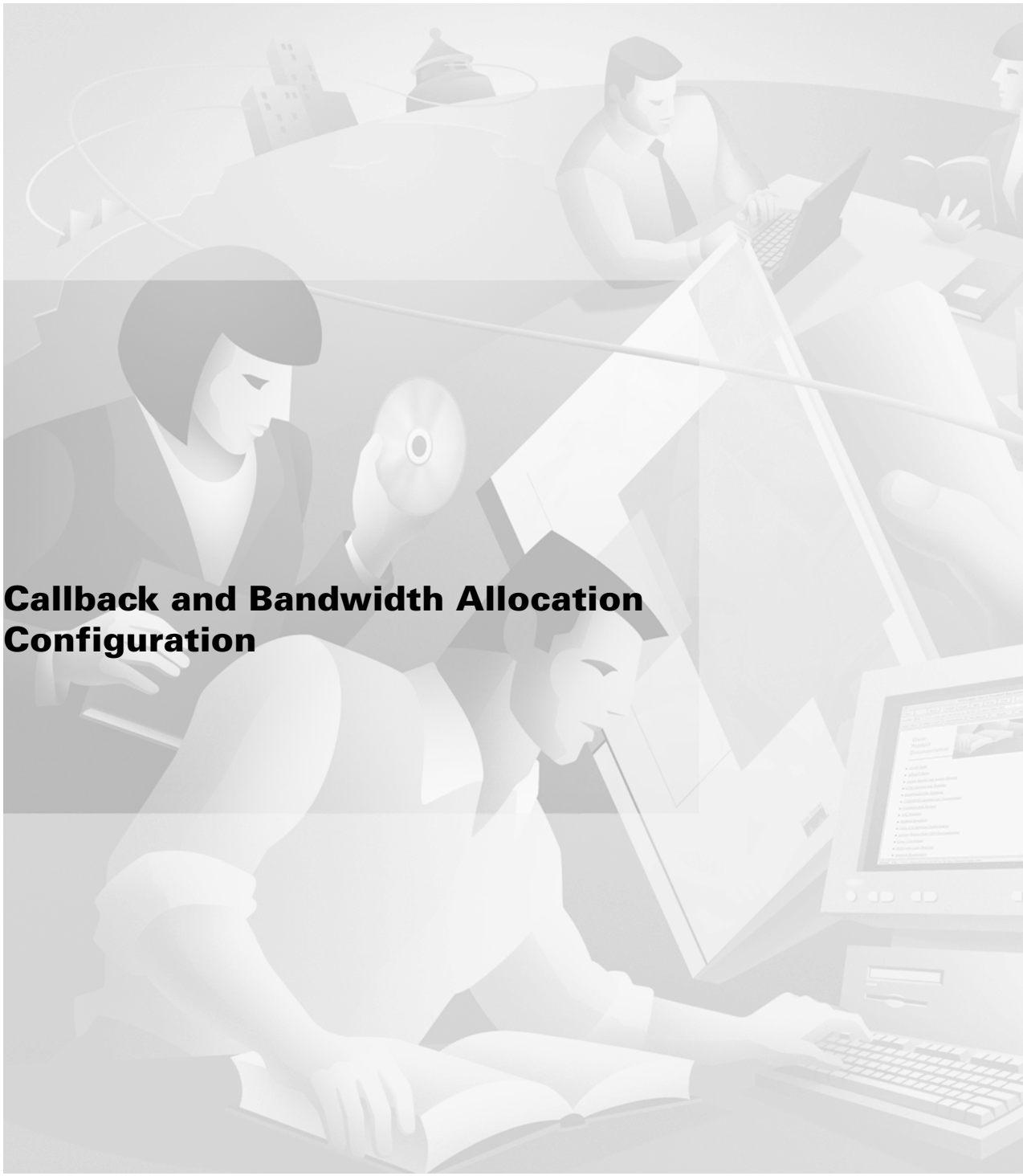
ISDN PRIs and BRIs by default are dialer interfaces. That is, a PRI configured without an explicit **interface dialer** command is *still* a dialer interface. The following example configures ISDN PRI. The D-channel configuration on serial interface 0:23 is applied to all the B channels. MMP is enabled, but no virtual interface template needs to be defined.

```
sgbp group stackq
sgbp member systemb 10.1.1.2
sgbp member systemc 10.1.1.3

username stackq password therock

isdn switch-type primary-4ess
controller t1 0
 framing esf
 linecode b8zs
 pri-group timeslots 1-23

isdn switch-type basic-net3
interface Serial0:23
ip unnumbered e0
 dialer map .....
 encap ppp
 ppp authentication chap
 dialer-group 1
 dialer rot 1
!
ppp multilink
```

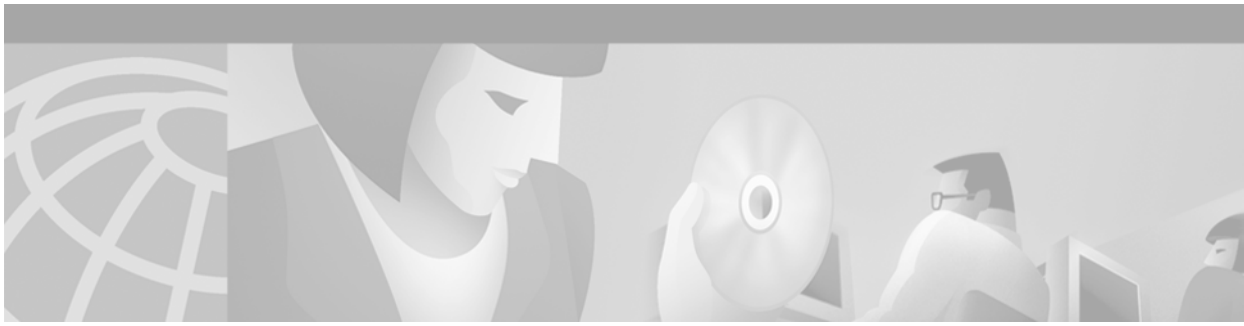
MMP with Offload Server

The following example shows a virtual template interface for a system that is being configured as an offload server (via the **sgbp seed-bid offload** command). All other stack group members must be defined with the **sgbp seed-bid default** command (or if you do not enter any **sgbp seed-bid** command, it defaults to this command).

```
multilink virtual-template 1
 sgbp group stackq
 sgbp member systemb 10.1.1.2
 sgbp member systemc 10.1.1.3
 sgbp seed-bid offload
 username stackq password therock

interface virtual-template 1
 ip unnumbered e0
 no ip route-cache
 ppp authentication chap
 ppp multilink
```



Callback and Bandwidth Allocation Configuration



Configuring Asynchronous Callback

This chapter describes how to configure Cisco IOS software to call back an asynchronous device that dials in, requests a callback from the router, and then disconnects. It includes the following main sections:

- Asynchronous Callback Overview
- How to Configure Asynchronous Callback
- Configuration Examples for Asynchronous Callback

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Asynchronous Callback Overview

Asynchronous callback is supported for the PPP and AppleTalk Remote Access (ARA) protocols. Callback is also supported on other interface types for PPP, including ISDN and any device that calls in and connects to the router at the EXEC level.

All callback sessions are returned on TTY lines. ARA is supported on virtual terminal lines, but also is supported on TTY lines if the **vty-arap** command is used. PPP, however, is supported on interfaces. Therefore, to enable PPP callback, you must enter the **autoselect ppp** command on the callback lines.

All current security mechanisms supported in Cisco IOS software are supported by the callback facility, including the following:

- TACACS+
- Challenge Handshake Authentication Protocol (CHAP) and Password Authentication Protocol (PAP) for PPP
- Per-user authentication for EXEC callback and ARA callback

The call originator must have the appropriate permissions set on the router before it can initiate a callback session.

Callback is useful for two purposes:

- Cost savings on toll calls

  For example, suppose it costs more to call from clients in Zone A to devices in Zone D than to call from Zone D to Zone A—costs are lower when devices in Zone D call back clients in Zone A.

- Consolidation and centralization of phone billing

  For example, if a corporation has 64 dial-in clients, enabling its routers to call back these clients consolidates billing. Instead of 64 phone bills, the corporation receives one bill.

How to Configure Asynchronous Callback

To configure asynchronous callback, perform the tasks in the following sections:

- Configuring Callback PPP Clients (Required)
- Enabling PPP Callback on Outgoing Lines (Required)
- Enabling Callback Clients That Dial In and Connect to the EXEC Prompt (Required)
- Configuring Callback ARA Clients (Required)

See the section "Configuration Examples for Asynchronous Callback" at the end of this chapter for ideas on how to implement asynchronous callback.

Configuring Callback PPP Clients

You can call back PPP clients that dial in to asynchronous interfaces. You can enable callback to the following two types of PPP clients:

- Clients that implement PPP callback per RFC 1570 (as an link control protocol, or LCP, negotiated extension).
- Clients that do not negotiate callback but can put themselves in answer-mode, whereby a callback from the router is accepted.

This section describes how to enable callback to each of these types of PPP clients.

Accepting Callback Requests from RFC-Compliant PPP Clients

To accept a callback request from an RFC 1570 PPP-compliant client, use the following command in interface (asynchronous) configuration mode:

Command	Purpose
`Router(config-if)# ppp callback accept`	Enables callback requests from RFC 1570 PPP-compliant clients on an asynchronous interface.

To configure Cisco IOS software to call back the originating PPP client, see the section "Enabling PPP Callback on Outgoing Lines" later in this chapter.

Accepting Callback Requests from Non-RFC-Compliant PPP Clients Placing Themselves in Answer Mode

A PPP client can put itself in answer-mode and can still be called back by the router, even though it cannot specifically request callback. To enable callback on the router to this type of client, use the following command in interface (asynchronous) configuration mode:

Command	Purpose
`Router(config-if)# ppp callback initiate`	Initiates callback requests from non-RFC 1570 PPP-compliant clients on an asynchronous interface.

To configure Cisco IOS software to call back the originating PPP client, see the next section, "Enabling PPP Callback on Outgoing Lines."

Enabling PPP Callback on Outgoing Lines

After enabling PPP clients to connect to an asynchronous interface and wait for a callback, you must place one or more TTY lines in PPP mode. Although calls from PPP clients enter through an asynchronous interface, the calls exit the client on a line placed in PPP mode.

To enable PPP client callback on outgoing TTY lines, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# chat-script script-name expect-send`	Defines a chat script to be applied when a PPP client requests callback.
Step 2	`Router(config)# username name [callback-dialstring telephone-number]`	Specifies a per-username callback dial string.
Step 3	`Router(config)# username name [callback-rotary rotary-group-number]`	Specifies a per-username rotary group for callback.
Step 4	`Router(config)# username name [callback-line [tty] line-number [ending-line-number]]`	Specifies a per-username line or set of lines for callback.
Step 5	`Router(config)# line [tty] line-number [ending-line-number]`	Enters line configuration mode.
Step 6	`Router(config-line)# autoselect ppp`	Configures automatic PPP startup on a line or set of lines.
Step 7	`Router(config-line)# login {authentication | local}`	Enables authentication on the line.
Step 8	`Router(config-line)# script callback regexp`	Applies a chat script to a line or set of lines.
Step 9	`Router(config-line)# callback forced-wait number-of-seconds`	Delays the callback for client modems that require a rest period before receiving a callback.

A client can issue a callback dial string; that dial string is used *only* if the dial string on the router is specified as NULL or is not defined. The recommended PPP chat script follows:

```
chat-script name ABORT ERROR ABORT BUSY "" "ATZ" OK "ATDT \T" TIMEOUT 30 CONNECT \c
```

See the section "Callback to a PPP Client Example" at the end of this chapter for a configuration example.

Note Normally a router avoids line and modem noise by clearing the initial data received within the first one or two seconds. However, when the autoselect PPP feature is configured, the router flushes characters initially received and then waits for more traffic. This flush causes time out problems with applications that send only one carriage return. To ensure that the input data sent by a modem or other asynchronous device is not lost after line activation, enter the **no flush-at-activation** line configuration command.

Enabling Callback Clients That Dial In and Connect to the EXEC Prompt

You can call back clients that dial in to a TTY line and connect to the EXEC prompt. To enable callback, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# service exec-callback`	Enables EXEC callback.
Step 2	`Router(config)# chat-script script-name expect-send`	Defines a chat script to be applied when clients dial in to the EXEC prompt.
Step 3	`Router(config)# username name [callback-dialstring telephone-number]`	Specifies a per-username callback dial string.
Step 4	`Router(config)# username name [callback-rotary rotary-group-number]`	Specifies a per-username rotary group for callback.
Step 5	`Router(config)# username name [callback-line [aux | tty] line-number [ending-line-number]]`	Specifies a per-username line or set of lines for callback.
Step 6	`Router(config)# username name [nocallback-verify]`	Does not require authentication on EXEC callback.
Step 7	`Router(config)# line [tty] line-number [ending-line-number]`	Enters line configuration mode.
Step 8	`Router(config-line)# script callback regexp`	Applies a chat script to the line or a set of lines.
Step 9	`Router(config-line)# callback forced-wait number-of-seconds`	Delays the callback for client modems that require a rest period before receiving a callback.

The recommended EXEC chat script follows:

```
chat-script name ABORT ERROR ABORT BUSY "" "ATZ" OK "ATDT \T" TIMEOUT 30 CONNECT \c
```

See the section "Callback Clients That Connect to the EXEC Prompt Example" at the end of this chapter for a configuration example.

Configuring Callback ARA Clients

To configure callback of ARA clients, use the following commands beginning in global configuration mode. These steps assume that you have already enabled AppleTalk routing and ARA.

	Command	Purpose
Step 1	`Router(config)# arap callback`	Enables callback to an ARA client.
Step 2	`Router(config)# chat-script script-name expect-send`	Defines a chat script to be applied when an ARA client connects to a TTY line and requests callback.
Step 3	`Router(config)# line [tty] line-number [ending-line-number]`	Enters line configuration mode.
Step 4	`Router(config-line)# arap enable`	Enables ARA on the line.
Step 5	`Router(config-line)# autoselect arap`	Configures automatic protocol startup on the line.
Step 6	`Router(config-line)# login {authentication | local}`	Enables authentication on the line.
Step 7	`Router(config-line)# script arap-callback regexp`	Applies an ARA-specific chat script to a line or set of lines.
Step 8	`Router(config-line)# callback forced-wait number-of-seconds`	Delays the callback for client modems that require a rest period before receiving a callback.
Step 9	`Router(config-line)# exit`	Returns to global configuration mode.
Step 10	`Router(config)# username name [callback-dialstring telephone-number]`	Specifies a per-username callback dial string.
Step 11	`Router(config)# username name [callback-rotary rotary-group-number]`	Specifies a per-username rotary group for callback.
Step 12	`Router(config)# username name [callback-line [tty] line-number [ending-line-number]]`	Specifies a per-username line or set of lines for callback.

The recommended ARA chat script follows and includes vendor-specific extensions on the Telebit 3000 modem to disable error control. Refer to the manual for your modem for the specific commands to disable error correction for ARA.

```
chat-script name ABORT ERROR ABORT BUSY "" "ATZ" OK "ATS180=0" OK "ATS181=1" OK "ATDT \T"
TIMEOUT 60 CONNECT \c
```

See the section "Callback to an ARA Client Example" at the end of this chapter for an example of calling back a PPP client.

Configuration Examples for Asynchronous Callback

The following sections provide asynchronous callback configuration examples:

- Callback to a PPP Client Example
- Callback Clients That Connect to the EXEC Prompt Example
- Callback to an ARA Client Example

Callback to a PPP Client Example

The following example shows the process of configuring callback to a PPP client on rotary 77. PAP authentication is enabled for PPP on the asynchronous interfaces. The **login local** command enables local username authentication on lines 7, 8, and 9. The remote PPP client host name is Ted, and the callback number is fixed at 1234567.

```
username Ted callback-dialstring "1234567" callback-rotary 77
            password Rhoda
interface async 7
 ip unnumbered ethernet 0
 encapsulation ppp
 no keepalive
 async default ip address 10.1.1.1
 async mode interactive
 ppp callback accept
 ppp authentication pap

interface async 8
 ip unnumbered ethernet 0
 encapsulation ppp
 no keepalive
 async default ip address 10.1.1.2
 async mode interactive
 ppp callback accept
 ppp authentication pap

interface async 9
 ip unnumbered ethernet 0
 encapsulation ppp
 no keepalive
 async default ip address 10.1.1.3
 async mode interactive
 ppp callback accept
 ppp authentication pap

line 7
 login local
 modem InOut
 rotary 77
 autoselect ppp

line 8
 login local
 modem InOut
 rotary 77
 autoselect ppp

line 9
 login local
 modem InOut
 rotary 77
 autoselect ppp
```

Callback Clients That Connect to the EXEC Prompt Example

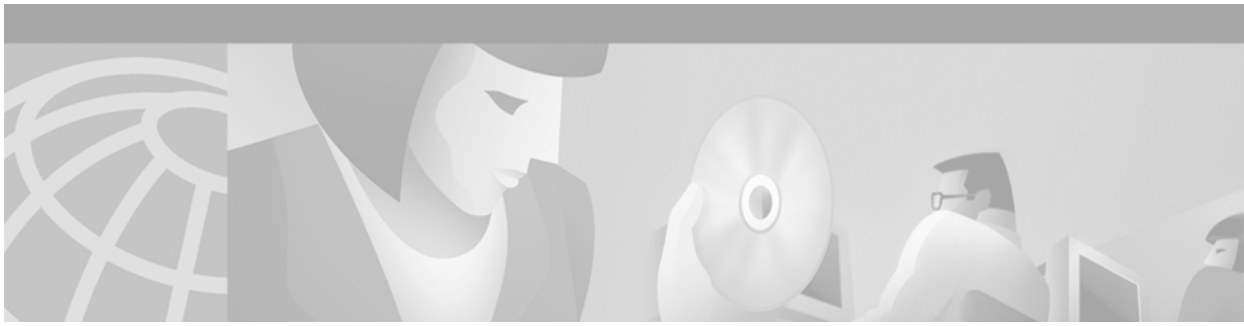
The following example shows the process to configure an outgoing callback on the same line as the incoming request. The **login local** command enables local username authentication on lines 4 and 7. Reauthentication is required upon reconnection.

```
service exec-callback
username milarepa callback-dialstring "" password letmein
line 4
 login local
line 7
 login local
```

Callback to an ARA Client Example

The following example shows the process of configuring callback to an ARA client on line 7. The **login local** command enables local username authentication on lines 4 and 7. Line 7 will always be used for ARA callback, whether the incoming call enters line 4, 7, or 8.

```
appletalk routing
arap callback
arap network 422 router test
username excalibur callback-dialstring "123456" callback-line 7 password guenivere
line 4
 login local
 modem InOut
 autoselect arap
 arap enable
line 7
 login local
 modem InOut
 autoselect arap
 arap enable
line 8
 login local
 modem InOut
 autoselect arap
 arap enable
```



Configuring PPP Callback

This chapter describes how to configure PPP callback for dial-on-demand routing (DDR). It includes the following main sections:

This feature implements the following callback specifications of RFC 1570:

- For the client—Option 0, location is determined by user authentication.
- For the server—Option 0, location is determined by user authentication; Option 1, dialing string; and Option 3, E.164 number.

Return calls are made through the same dialer rotary group but not necessarily the same line as the initial call.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the PPP callback commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

PPP Callback for DDR Overview

PPP callback provides a client/server relationship between the endpoints of a point-to-point connection. PPP callback allows a router to request that a dialup peer router call back. The callback feature can be used to control access and toll costs between the routers.

When PPP callback is configured on the participating routers, the calling router (the callback client) passes authentication information to the remote router (the callback server), which uses the host name and dial string authentication information to determine whether to place a return call. If the authentication is successful, the callback server disconnects and then places a return call. The remote username of the return call is used to associate it with the initial call so that packets can be sent.

Both routers on a point-to-point link must be configured for PPP callback; one must function as a callback client and one must be configured as a callback server. The callback client must be configured to initiate PPP callback requests, and the callback server must be configured to accept PPP callback requests and place return calls.

See the section "MS Callback Overview" later in this chapter if you are using PPP callback between a Cisco router or access server and client devices configured for Windows 95 and Windows NT.

Note If the return call fails (because the line is not answered or the line is busy), no retry occurs. If the callback server has no interface available when attempting the return call, it does not retry.

How to Configure PPP Callback for DDR

To configure PPP callback for DDR, perform the following tasks:

- Configuring a Router as a Callback Client (Required)
- Configuring a Router as a Callback Server (Required)

For an example of configuring PPP callback, see the section "Configuration Examples for PPP Callback" at the end of this chapter.

Configuring a Router as a Callback Client

To configure a router interface as a callback client, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and enters interface configuration mode.
Step 2	`Router(config-if)# dialer in-band [no-parity | odd-parity]`	Enables DDR. Specifies parity, if needed, on synchronous or asynchronous serial interfaces.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp authentication {chap | pap}`	Enables CHAP or PAP authentication.
Step 5	`Router(config-if)# dialer map protocol next-hop-address name hostname dial-string`	Maps the next hop address to the host name and phone number.
Step 6	`Router(config-if)# ppp callback request`	Enables the interface to request PPP callback for this callback map class.
Step 7	`Router(config-if)# dialer hold-queue packets timeout seconds`	(Optional) Configures a dialer hold queue to store packets for this callback map class.

Configuring a Router as a Callback Server

To configure a router as a callback server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and enters interface configuration mode.
Step 2	`Router(config-if)# dialer in-band [no-parity | odd-parity]`	Enables DDR. Specifies parity, if needed, on synchronous or asynchronous serial interfaces.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp authentication {chap | pap}`	Enables CHAP or PAP authentication.
Step 5	`Router(config-if)# dialer map protocol next-hop-address name hostname class classname dial-string`	Maps the next hop address to the host name and phone number, using the name of the map class established for PPP callback on this interface.
Step 6	`Router(config-if)# dialer hold-queue number timeout seconds`	(Optional) Configures a dialer hold queue to store packets to be transferred when the callback connection is established.
Step 7	`Router(config-if)# dialer enable-timeout seconds`	(Optional) Configures a timeout period between calls.
Step 8	`Router(config-if)# ppp callback accept`	Configures the interface to accept PPP callback.
Step 9	`Router(config-if)# isdn fast-rollover-delay seconds`	(ISDN only) Configures the time to wait before another call is placed on a B channel to allow the prior call to be torn down completely.
Step 10	`Router(config-if)# dialer callback-secure`	(Optional) Enables callback security, if desired.
Step 11	`Router(config-if)# exit`	Returns to global configuration mode.
Step 12	`Router(config-map-class)# map-class dialer classname`	Configures a dialer map class for PPP callback.
Step 13	`Router(config-map-class)# dialer callback-server [username]`	Configures a dialer map class as a callback server.

Note On the PPP callback server, the **dialer enable-timeout** command functions as the timer for returning calls to the callback client.

MS Callback Overview

MS Callback provides client/server callback services for Microsoft Windows 95 and Microsoft Windows NT clients. MS Callback supports the Microsoft Callback Control Protocol (MSCB). MSCB is a Microsoft proprietary protocol that is used by Windows 95 and Windows NT clients. MS Callback supports negotiated PPP Link Control Protocol (LCP) extensions initiated and agreed upon by the Microsoft client. The MS Callback feature is added to existing PPP Callback functionality. Therefore, if you configure your Cisco access server to perform PPP Callback using Cisco IOS Release 11.3(2)T or later, MS Callback is automatically available.

MS Callback supports authentication, authorization, and accounting (AAA) security models using a local database or AAA server.

MSCB uses LCP callback options with suboption type 6. The Cisco MS Callback feature supports clients with a user-specified callback number and server specified (preconfigured) callback number.

MS Callback does not affect non-Microsoft machines that implement standard PPP LCP extensions as described in RFC 1570. In this scenario, MS Callback is transparent.

The following are restrictions of the MS Callback feature:

- The Cisco access server and client must be configured for PPP and PPP callback.
- The router or access server must be configured to use CHAP or PAP authorization.
- MS Callback is only supported on the Public Switched Telephone Network (PSTN) and ISDN links.
- MS Callback is only supported for IP.

How to Configure MS Callback

If you configure the Cisco access server for PPP callback, MS Callback is enabled by default. You need not configure additional parameters on the Cisco access server. If an interface is configured to accept PPP callbacks, and a client attempts to cancel the callback, Cisco IOS software will refuse the request and disconnect the client. If a client is allowed to cancel callbacks, the **ppp callback permit** command must be configured on the interface.

To debug PPP connections using MS Callback, see the **debug ppp cbcp** command in the *Cisco IOS Debug Command Reference* publication.

For more information on configuring MS Callback, see the following URL.

http://www.cisco.com/en/US/customer/tech/tk801/tk36/technologies_configuration_example09186a0080094338.shtml

Configuration Examples for PPP Callback

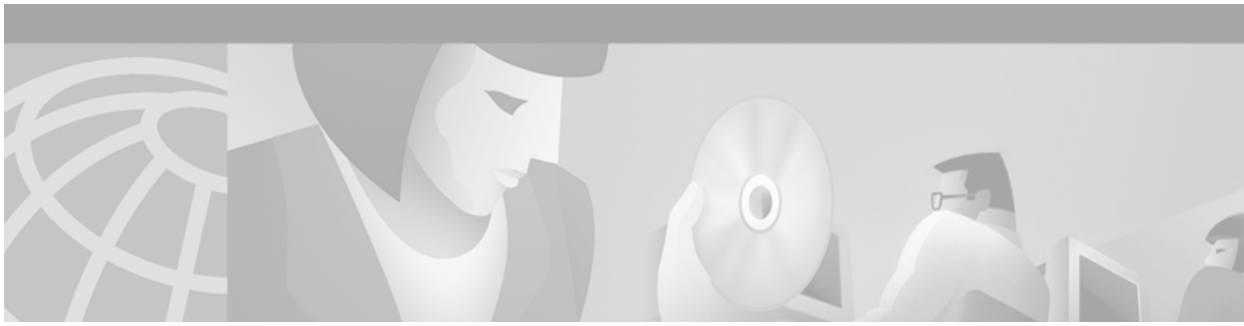
The following example configures a PPP callback server and client to call each other. The PPP callback server is configured on an ISDN BRI interface in a router in Atlanta. The callback server requires an enable timeout and a map class to be defined. The PPP callback client is configured on an ISDN BRI interface in a router in Dallas. The callback client does not require an enable timeout and a map class to be defined. The **dialer map** command is not required on the Cisco access server when MS Callback is enabled.

PPP Callback Server

```
interface bri 0
 ip address 10.1.1.7 255.255.255.0
 encapsulation ppp
 dialer callback-secure
 dialer enable-timeout 2
 dialer map ip 10.1.1.8 name class1 class dial1 81012345678901
 dialer-group 1
 ppp callback accept
 ppp authentication chap
!
map-class dialer dial1
 dialer callback-server user1
```

PPP Callback Client

```
interface bri 0
 ip address 10.1.1.8 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.7 name class2 81012345678902
 dialer-group 1
 ppp callback request
 ppp authentication chap
```



Configuring ISDN Caller ID Callback

This chapter describes how to configure the ISDN Caller ID Callback feature. It includes the following main sections:

The ISDN Caller ID Callback feature conflicts with dialer callback security inherent in the dialer profiles feature for dial-on-demand routing (DDR). If dialer callback security is configured, it takes precedence; ISDN caller ID callback is ignored.

Caller ID screening requires a local switch that is capable of delivering the caller ID to the router or access server. If you enable caller ID screening but do not have such a switch, no calls will be allowed in.

ISDN caller ID callback requires DDR to be configured and bidirectional dialing to be working between the calling and callback routers. Detailed DDR prerequisites depend on whether you have configured legacy DDR or dialer profiles.

For a legacy DDR configuration, ISDN caller ID callback has the following prerequisite:

- A **dialer map** command is configured for the dial string that is used in the incoming call setup message. The dial string is used in the callback.

For a dialer profiles configuration, ISDN caller ID callback has the following prerequisites:

- A **dialer caller** command is configured to screen for the dial-in number.
- A **dialer string** command is configured with the number to use in the callback.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the ISDN caller ID callback commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

ISDN Caller ID Callback Overview

ISDN caller ID callback allows the initial incoming call from the client to the server to be rejected on the basis of the caller ID message contained in the ISDN setup message, and it allows a callback to be initiated to the calling destination.

Before Cisco IOS Release 11.2 F, ISDN callback functionality required PPP or Combinet Packet Protocol (CPP) client authentication and client/server callback negotiation to proceed. If authentication and callback negotiation were successful, the callback server had to disconnect the call and then place a return call. Both the initial call and the return call were subject to tolls, and when service providers charge by the minute, even brief calls could be expensive.

This feature is independent of the encapsulation in effect and can be used with various encapsulations, such as PPP, High-Level Data Link Control (HDLC), Frame Relay, and X.25.

The ISDN Caller ID Callback feature allows users to control costs because charges do not apply to the initial, rejected call.

ISDN caller ID callback allows great flexibility for you to define which calls to accept, which to deny, and which calls to reject initially but for which the router should initiate callback. The feature works by using existing ISDN caller ID screening, which matches the number in the incoming call against numbers configured on the router, determining the best match for the number in the incoming call, and then, if configured, initiating callback to the number configured on the router.

When a call is received, the entire list of configured numbers is checked and the configuration of the best match number determines the action:

- If the incoming number is best matched by a number that is configured for callback, the incoming call is rejected and callback is initiated.
- If the incoming number is best matched by another entry in the list of configured numbers, the call is accepted.
- If the incoming number does not match any entry in the configured list, the call is rejected and no callback is started.

“Don’t care” characters are allowed in the caller ID screening configuration on the router and are used to determine the best match.

For more information and examples, see the “Best Match System Examples” section later in this document.

Callback After the Best Match Is Determined

The details of router activities after the router finds a best match with callback depend on the DDR feature that is configured. The ISDN Caller ID Callback feature works with the following DDR features:

- Legacy DDR
- Dialer Profiles

Legacy DDR

If legacy DDR is configured for the host or user that is identified in the incoming call message, the router performs the following actions:

1. Checks the table of configured numbers for caller ID callback.
2. Searches the **dialer map** entries for a number that “best matches” the incoming call string.

3. Waits for a configured length of time to expire.
4. Initiates callback to the number provided in the **dialer map** command.

Dialer Profiles

If the dialer profiles are configured for the host or user identified in the incoming call message, the router performs the following actions:

1. Searches through all the dialer pool members to match the incoming call number to a **dialer caller** number.
2. Initiates a callback to the dialer profile.
3. Waits for a configured length of time to expire.
4. Calls the number identified in the **dialer string** command associated with the dialer profile.

Timing and Coordinating Callback on Both Sides

When an incoming call arrives and the router finds a best match configured for callback, the router uses the value configured by the **dialer enable-timeout** command to determine the length of time to wait before making the callback.

The minimum value of the timer is 1 second; the default value of the timer is 15 seconds. The interval set for this feature on the router must be much less than that set for DDR fast call rerouting for ISDN (that interval is set by the **dialer wait-for-carrier-time** command) on the calling (remote) side. We recommend setting the dialer wait-for-carrier timer on the calling side to twice the length of the dialer enable-timeout timer on the callback side.

Note The remote site cannot be configured for multiple dial-in numbers because a busy callback number or a rejected call causes the second number to be tried. That number might be located at a different site, defeating the purpose of the callback.

How to Configure ISDN Caller ID Callback

To configure ISDN caller ID callback, perform the tasks in the following sections. The required configuration tasks depend whether you have configured legacy DDR or dialer profiles.

- Configuring ISDN Caller ID Callback for Legacy DDR (As required)
- Configuring ISDN Caller ID Callback for Dialer Profiles (As required)

For configuration examples, see the section "Configuration Examples for ISDN Caller ID Callback" at the end of this chapter.

Configuring ISDN Caller ID Callback for Legacy DDR

This section provides configuration tasks for the local (server, callback) side and the remote (client, calling) side.

On the callback (local) side, to configure ISDN caller ID callback when legacy DDR is configured, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# isdn caller remote-number callback`	Configures caller ID screening and callback when a dialer rotary is not configured.
	or	
	`Router(config-if)# dialer caller number callback`	Configures caller ID screening and callback when a dialer rotary (dialer interface) is configured.
Step 2	`Router(config-if)# dialer enable-timeout seconds`	Configures the time to wait before initiating callback.

On the calling (remote) side, to set the timer for fast call rerouting, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer wait-for-carrier-time seconds`	Changes the ISDN fast call rerouting timer to double the length of the enable timeout timer.

Configuring ISDN Caller ID Callback for Dialer Profiles

This section provides configuration tasks for the local side and the remote side.

On the callback (local) side, to configure ISDN caller ID callback when the dialer profiles are configured, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# dialer caller number callback`	Configures caller ID screening and callback.
Step 2	`Router(config-if)# dialer enable-timeout seconds`	Configures the time to wait before initiating callback.

On the calling (remote) side, to set the timer for fast call rerouting, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer wait-for-carrier-time seconds`	Changes the ISDN fast call rerouting timer to double the length of the enable timeout timer.

Monitoring and Troubleshooting ISDN Caller ID Callback

To monitor and troubleshoot ISDN caller ID callback, use the following commands in EXEC mode as needed:

Command	Purpose
`Router# show dialer`	Displays information about the status and configuration of the ISDN interface on the router.
`Router# debug isdn event`	Displays ISDN 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 tear down of ISDN network connections).
`Router# debug isdn q931`	Displays Layer 3 signaling messages, protocol transitions and processes, the line protocol state, and the channel IDs for each ISDN interface.

Configuration Examples for ISDN Caller ID Callback

The following sections provide ISDN caller ID callback configuration examples:

- Best Match System Examples
- Simple Callback Configuration Examples
- ISDN Caller ID Callback with Dialer Profiles Examples
- ISDN Caller ID Callback with Legacy DDR Example

Best Match System Examples

The best match is determined by matching the incoming number against the numbers in the configured callback commands, starting with the right-most character in the numbers and using the letter X for any “don’t care” characters in the configured commands. If multiple configured numbers match an incoming number, the best match is the one with the fewest “don’t care” characters.

The reason for using a system based on right-most matching is that a given number can be represented in many different ways. For example, all the following items might be used to represent the same number, depending on the circumstances (international call, long-distance domestic call, call through a PBX, and so forth):

```
011 1 408 555 7654
    1 408 555 7654
      408 555 7654
          555 7654
            5 7654
```

Best Match Based on the Number of "Don't Care" Characters Example

The following example assumes that you have an incoming call from one of the numbers from the previous example entered (4085557654), and that you configured the following numbers for callback on the router (disregarding for the moment the commands that can be used to configure callback):

```
555xxxx callback
5552xxx callback
555865x
5554654 callback
xxxxx
```

The first number listed is the best match for the incoming number (in the configured number, the three numbers and four Xs all match the incoming number); the line indicates that callback is to be initiated. The last line has five Xs; it is not the best match for the calling number.

Note The last number in the list shown allows calls from any other number to be accepted without callback. When you use such a line, you must make sure that the number of Xs in the line exceeds the number of Xs in any other line. In the last line, five Xs are used; the other lines use at most four Xs.

The order of configured numbers is not important; the router searches the entire list and then determines the best match.

Best Match with No Callback Configured Example

The following example assumes that a call comes from the same number (4085557654) and that only the following numbers are configured:

```
5552xxx callback
555865x
5554654 callback
xxxxx
```

In this case, the best match is in the final line listed, so the incoming call is accepted but callback is not initiated.

No Match Configured Example

The following example assumes that a call comes from the same number (4085557654) and that only the following numbers are configured:

```
5552xxx callback
555865x
5554654 callback
```

In this case, there is no match at all, and the call is just rejected.

Simple Callback Configuration Examples

The following example assumes that callback calls will be made only to numbers in the 555 and 556 exchanges but that any other number can call in:

```
isdn caller 408555xxxx callback
isdn caller 408556xxxx callback
isdn caller xxxxx
```

The following example configures the router to accept a call with a delivered caller ID equal to 4155551234:

```
isdn caller 4155551234
```

The following example configures the router to accept a call with a delivered caller ID equal to 41555512 with any digits in the last two positions:

```
isdn caller 41555512xx
```

The following example configures the router to make a callback to a delivered caller ID equal to 41555512 with any digits in the last two positions. (The router rejects the call initially, and then makes the callback.) The router accepts calls from any other numbers.

```
isdn caller 41555512xx callback
isdn caller xxx
```

ISDN Caller ID Callback with Dialer Profiles Examples

The following example shows the configuration of a central site that can place or receive calls from three remote sites over four ISDN BRI lines. Each remote site is on a different IP subnet and has different bandwidth requirements. Therefore, three dialer interfaces and three dialer pools are defined.

```
! This is a dialer profile for reaching remote subnetwork 10.1.1.1.
interface dialer 1
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer remote-name Smalluser
 dialer string 4540
 dialer pool 3
 dialer-group 1
 dialer caller 14802616900 callback
 dialer caller 1480262xxxx callback
!
! This is a dialer profile for reaching remote subnetwork 10.2.2.2.
interface dialer 2
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name Mediumuser
 dialer string 5264540 class Eng
 dialer load-threshold 50 either
 dialer pool 1
 dialer-group 2
 dialer caller 14805364540 callback
 dialer caller 1480267xxxx callback
 dialer enable-timeout 2
!
! This is a dialer profile for reaching remote subnetwork 10.3.3.3.
interface dialer 3
 ip address 10.3.3.3 255.255.255.0
 encapsulation ppp
 dialer remote-name Poweruser
 dialer string 4156884540 class Eng
 dialer hold-queue 10
 dialer load-threshold 80
 dialer pool 2
 dialer-group 2
!
! This map class ensures that these calls use an ISDN speed of 56 kbps.
map-class dialer Eng
 isdn speed 56
```

```
!
interface bri 0
 encapsulation PPP
! BRI 0 has a higher priority than BRI 1 in dialer pool 1.
 dialer pool-member 1 priority 100
 ppp authentication chap
!
interface bri 1
 encapsulation ppp
 dialer pool-member 1 priority 50
 dialer pool-member 2 priority 50
! BRI 1 has a reserved channel in dialer pool 3; the channel remains inactive
! until BRI 1 uses it to place calls.
 dialer pool-member 3 min-link 1
 ppp authentication chap
!
interface bri 2
 encapsulation ppp
! BRI 2 has a higher priority than BRI 1 in dialer pool 2.
dialer pool-member 2 priority 100
ppp authentication chap
!
interface bri 3
encapsulation ppp
! BRI 3 has the highest priority in dialer pool 2.
dialer pool-member 2 priority 150
ppp authentication chap
```

ISDN Caller ID Callback with Legacy DDR Example

This section provides two examples of caller ID callback with legacy DDR:

- Individual Interface Example
- Dialer Rotary Group Example

Individual Interface Example

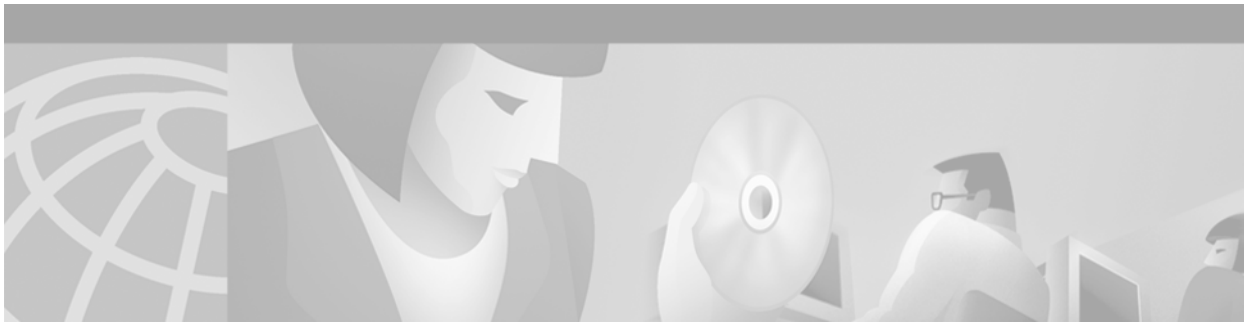
The following example configures a BRI interface for legacy DDR and ISDN caller ID callback:

```
interface bri 0
 description Connected to NTT 81012345678901
 ip address 10.1.1.7 255.255.255.0
 no ip mroute-cache
 encapsulation ppp
 isdn caller 81012345678902 callback
 dialer enable-timeout 2
 dialer map ip 10.1.1.8 name spanky 81012345678902
 dialer-group 1
 ppp authentication chap
```

Dialer Rotary Group Example

The following example configures BRI interfaces to connect into a rotary group (dialer group) and then configures a dialer interface for that dialer group. This configuration permits IP packets to trigger calls. The dialer interface is configured to initiate callback to any number in the 1-480-261 exchange and to accept calls from two other specific numbers.

```
interface bri 0
 description connected into a rotary group
 encapsulation ppp
 dialer rotary-group 1
!
interface bri 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface bri 2
 encapsulation ppp
 dialer rotary-group 1
!
interface bri 3
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface bri 4
 encapsulation ppp
 dialer rotary-group 1
!
interface dialer 1
 description Dialer group controlling the BRIs
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name angus 14802616900
 dialer map ip 10.1.1.3 name shamus 14802616901
 dialer map ip 10.1.1.4 name larry 14807362060
 dialer map ip 10.1.1.5 name wally 19165561424
 dialer map ip 10.1.1.6 name shemp 12129767448
 dialer-group 1
 ppp authentication chap
!
 dialer caller 1480261xxxx callback
 dialer caller 19165561424
 dialer caller 12129767448
!
dialer-list 1 protocol ip permit
```



Configuring BACP

This chapter describes how to configure the Bandwidth Allocation Control Protocol (BACP), described in RFC 2125. It includes the following main sections:

- BACP Overview
- How to Configure BACP
- Monitoring and Maintaining Interfaces Configured for BACP
- Troubleshooting BACP
- Configuration Examples for BACP

BACP requires a system only to have the knowledge of its own phone numbers and link types. A system must be able to provide the phone numbers and link type to its peer to satisfy the call control mechanism. (Certain situations might not be able to satisfy this requirement; numbers might not be present because of security considerations.)

BACP is designed to operate in both the virtual interface environment and the dialer interface environment. It can operate over any physical interface that is Multilink PPP-capable and has a dial capability; at initial release, BACP supports ISDN and asynchronous serial interfaces.

The addition of any link to an existing multilink bundle is controlled by a Bandwidth Allocation Protocol (BAP) call or callback request message, and the removal of a link can be controlled by a link drop message.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the PPP BACP commands in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

BACP Overview

The BACP provides Multilink PPP (MLP) peers with the ability to govern link utilization. Once peers have successfully negotiated BACP, they can use the BAP, which is a subset of BACP, to negotiate bandwidth allocation. BAP provides a set of rules governing dynamic bandwidth allocation through call control; a defined method for adding and removing links from a multilink bundle for Multilink PPP is used.

BACP provides the following benefits:

- Allows multilink implementations to interoperate by providing call control through the use of link types, speeds, and telephone numbers.
- Controls thrashing caused by links being brought up and removed in a short period of time.
- Ensures that both ends of the link are informed when links are added or removed from a multilink bundle.

For simplicity, the remaining text of this chapter makes no distinction between BACP and BAP; only BACP is mentioned.

BACP Configuration Options

PPP BACP can be configured to operate in the following ways:

- Passive mode (default)—The system accepts incoming calls; the calls might request callback, addition of a link, or removal of a link from a multilink bundle. The system also monitors the multilink load by default.

  Passive mode is for virtual template interfaces or for dialer interfaces.

- Active mode—The system initiates outbound calls, sets the parameters for outbound calls, and determines whether links should be added to or removed from a multilink bundle. The system also monitors the multilink load by default.

  Active mode is for dialer interfaces, but not for virtual template interfaces. (If you attempt to configure active mode on a virtual template interface, no calls will be made.)

A virtual or dialer interface must be configured either to make call requests or to make callback requests, but it cannot be configured to do both.

Support of BACP on virtual interfaces in an Multichassis Multilink PPP (MMP) environment is restricted to incoming calls on the multilink group. Support of BACP for outgoing calls is provided by dialer interface configuration only.

BACP supports only ISDN and asynchronous serial interfaces.

Dialer support is provided only for legacy dial-on-demand routing (DDR) dialer configurations; BACP cannot be used in conjunction with the DDR dialer profiles feature.

BACP is configured on virtual template interfaces and physical interfaces that are multilink capable. For both the virtual template interfaces and the dialer interfaces, BACP requires MMP and bidirectional dialing to be working between the routers that will negotiate control and allocation of bandwidth for the multilink bundle.

How to Configure BACP

Before you configure BACP on an interface, determine the following important information. The router might be unable to connect to a peer if this information is incorrect.

- Type of link (ISDN or analog) to be used. Link types must match on the local and remote ends of the link.
- Line speed needed to reach the remote peer. The speed configured for the local physical interface must be at least that of the link. The **bandwidth** command or the **dialer map** command with the **speed** keyword can be used.
- Local telephone number to be used for incoming PPP BACP calls, if it is different from a rotary group base number or if incoming PPP BACP calls should be directed to a specific number.

  During negotiations with a peer, PPP BACP might respond with a telephone number *delta*, indicating that the peer should modify certain digits of the dialed phone number and dial again to reach the PPP BACP interface or to set up another link.

BACP can be configured on a virtual template interface or on a dialer interface (including dialer rotary groups and ISDN interfaces).

To configure BACP on a selected interface or interface template, perform the following tasks in the order listed:

- Enabling BACP (Required)

  Passive mode is in effect and the values of several parameters are set by default when PPP BACP is enabled. If you can accept *all* the passive mode parameters, do not continue with the tasks.
- Modifying BACP Passive Mode Default Settings (As required)

  or
- Configuring Active Mode BACP (As required)

Note You can configure one interface in passive mode and another in active mode so that one interface accepts incoming call requests and makes callback requests (passive mode), and the other interface makes call requests and accepts callback requests (active mode).

A dialer or virtual template interface should be configured to reflect the required dial capability of the interface. A dial-in pool (in passive mode) might have no requirement to dial out but might want remote users to add multiple links, with the remote user incurring the cost of the call. Similarly, a dial-out configuration (active mode) suggests that the router is a client, rather than a server, on that link. The active-mode user incurs the cost of additional links.

You might need to configure a base telephone number, if it is applicable to your dial-in environment. This number is one that remote users can dial to establish a connection. Otherwise, individual PPP BACP links might need numbers. Information is provided in the task lists for configuring passive mode or active mode PPP BACP. See the **ppp bap number** command options in the task lists.

You can also troubleshoot BACP configuration and operations and monitor interfaces configured for PPP BACP. For details, see the "Troubleshooting BACP" and "Monitoring and Maintaining Interfaces Configured for BACP" sections later in this chapter.

See the section "Configuration Examples for BACP" at the end of this chapter for examples of PPP BACP configuration.

Enabling BACP

To enable PPP bandwidth allocation control and dynamic allocation of bandwidth, use one of the following commands in interface configuration mode:

Command	Purpose

`Router(config-if)# ppp multilink bap`
or
`Router(config-if)# ppp multilink bap required`
	Enables PPP BACP bandwidth allocation negotiation.
	
	Enables PPP BACP bandwidth allocation negotiation and enforces mandatory negotiation of BACP for the multilink bundle.

When PPP BACP is enabled, it is in passive mode by default and the following settings are in effect:

- Allows a peer to initiate link addition.
- Allows a peer to initiate link removal.
- Requests that a peer initiate link addition.
- Waits 20 seconds before timing out on pending actions.
- Waits 3 seconds before timing out on not receiving a response from a peer.
- Makes only one attempt to call a number.
- Makes up to three retries for sending a request.
- Searches for and logs up to five free dialers.
- Makes three attempts to send a call status indication.
- Adds only ISDN links to a multilink bundle.
- Monitors load.

The default settings will be in effect in the environment for which the **ppp multilink bap** command is entered:

- Virtual template interface, if that is where the command is entered.

  When the command is entered in a virtual template interface, configuration applies to any virtual access interface that is created dynamically under Multilink PPP, the application that defines the template.

- Dialer interface, if that is where the command is entered.

See the section "Basic BACP Configurations" at the end of this chapter for an example of how to configure BACP.

Modifying BACP Passive Mode Default Settings

To modify the default parameter values or to configure additional parameters in passive mode, use the following commands, as needed, in interface configuration mode for the interface or virtual template interface that is configured for PPP BACP:

Command	Purpose
`Router(config-if)# ppp bap timeout pending` *seconds*	Modifies the timeout on pending actions.
`Router(config-if)# ppp bap timeout response` *seconds*	Modifies the timeout on not receiving a response from a peer.
`Router(config-if)# ppp bap max dial-attempts` *number*	Modifies the number of attempts to call a number.
`Router(config-if)# ppp bap max ind-retries` *number*	Modifies the number of times to send a call status indication.
`Router(config-if)# ppp bap max req-retries` *number*	Modifies the number of retries of a particular request.
`Router(config-if)# ppp bap max dialers` *number*	Modifies the maximum number of free dialers logged.

`Router(config-if)# ppp bap link types analog`
 	
or
 	
`Router(config-if)# ppp bap link types isdn analog`
	Specifies that only analog links can be added to a multilink bundle.
	
	Allows both ISDN and analog links to be added.

`Router(config-if)# ppp bap number default` *phone-number*	For all DDR-capable interfaces in the group, specifies a primary telephone number for the peer to call for PPP BACP negotiation, if different from any base number defined on the dialer interface or virtual template interface.
`Router(config-if)# ppp bap number secondary` *phone-number*	For BRI interfaces on which a different number is provided for each B channel, specifies the secondary telephone number.
`Router(config-if)# ppp bap drop timer` *seconds*	Specifies a time to wait between outgoing link drop requests.
`Router(config-if)# no ppp bap monitor load`	Disables the default monitoring of load and the validation of peer requests against load thresholds.

See the section "Passive Mode Dialer Rotary Group Members with One Dial-In Number" later in this chapter for an example of how to configure passive mode parameters.

Configuring Active Mode BACP

To configure active mode BACP, use the following commands in interface configuration mode for the dialer interface on which BACP was enabled. For your convenience, the commands that make BACP function in active mode are presented before the commands that change default parameters or add parameters.

Command	Purpose
`Router(config-if)# ppp bap call request`	Enables the interface to initiate the addition of links to the multilink bundle.
`Router(config-if)# ppp bap callback accept`	Enables the interface to initiate the addition of links upon peer request.

Command	Purpose
`Router(config-if)# ppp bap drop after-retries`	Enables the interface to drop a link without negotiation after receiving no response to retries to send a drop request.
`Router(config-if)# ppp bap call timer` *seconds*	Sets the time to wait between outgoing call requests.
`Router(config-if)# ppp bap timeout pending` *seconds*	Modifies the timeout on pending actions.
`Router(config-if)# ppp bap timeout response` *seconds*	Modifies the timeout on not receiving a response from a peer.
`Router(config-if)# ppp bap max dial-attempts` *number*	Modifies the number of attempts to call a number.
`Router(config-if)# ppp bap max ind-retries` *number*	Modifies the number of times to send a call status indication.
`Router(config-if)# ppp bap max req-retries` *number*	Modifies the number of retries of a particular request.
`Router(config-if)# ppp bap max dialers` *number*	Modifies the maximum number of free dialers logged.

`Router(config-if)# ppp bap link types analog`
or
`Router(config-if)# ppp bap link types isdn analog`
	Specifies that only analog links can be added to a multilink bundle.
	
	Allows both ISDN and analog links to be added.

`Router(config-if)# ppp bap number default` *phone-number*	For all DDR-capable interfaces in the group, specifies a primary telephone number for the peer to call for PPP BACP negotiation, if different from any base number defined on the dialer interface or virtual template interface.
`Router(config-if)# ppp bap number secondary` *phone-number*	For BRI interfaces on which a different number is provided for each B channel, specifies the secondary telephone number.

When BACP is enabled, multiple dialer maps to one destination are not needed when they differ only by number. That is, once the initial call has been made to create the bundle, further dialing attempts are realized through the BACP phone number negotiation.

Outgoing calls are supported through the use of dialer maps. However, when an initial incoming call creates a dynamic dialer map, the router can dial out if the peer supplies a phone number. This capability is achieved by the dynamic creation of static dialer maps for BACP. These temporary dialer maps can be displayed by using the **show dialer map** command. These temporary dialer maps last only as long as the BACP group lasts and are removed when the BACP group or the associated map is removed.

Monitoring and Maintaining Interfaces Configured for BACP

To monitor interfaces configured for PPP BACP, use any of the following commands in EXEC mode:

Command	Purpose
`Router> show ppp bap group` [*name*]	Displays information about all PPP BACP multilink bundle groups or a specific, named multilink bundle group.
`Router> show ppp bap queues`	Displays information about the BACP queues.
`Router> show ppp multilink`	Displays information about the dialer interface, the multilink bundle, and the group members.
`Router> show dialer`	Displays BACP numbers dialed and the reasons for the calls.
`Router> show dialer map`	Displays configured dynamic and static dialer maps and dynamically created BACP temporary static dialer maps.

Troubleshooting BACP

To troubleshoot the BACP configuration and operation, use the following **debug** commands:

Command	Purpose
`Router> debug ppp bap [error | event | negotiation]`	Displays BACP errors, protocol actions, and negotiation events and transitions.
`Router> debug ppp multilink events`	Displays information about events affecting multilink bundles established for BACP.

Configuration Examples for BACP

The following sections provide BACP configuration examples:

- Basic BACP Configurations
- Dialer Rotary Group with Different Dial-In Numbers
- Passive Mode Dialer Rotary Group Members with One Dial-In Number
- PRI Interface with No Defined PPP BACP Number
- BRI Interface with No Defined BACP Number

Basic BACP Configurations

The following example configures an ISDN BRI interface for BACP to make outgoing calls and prevent the peer from negotiating link drops:

```
interface bri 0
 ip unnumbered ethernet 0
 dialer load-threshold 10 either
 dialer map ip 172.21.13.101 name bap-peer 12345668899
 encapsulation ppp
 ppp multilink bap
 ppp bap call request
 ppp bap callback accept
 no ppp bap call accept
 no ppp bap drop accept
 ppp bap pending timeout 30
 ppp bap number default 5664567
 ppp bap number secondary 5664568
```

The following example configures a dialer rotary group to accept incoming calls:

```
interface async 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
 ppp bap number default 5663456
!
! Set the bandwidth to suit the modem/line speed on the remote side.
interface bri 0
 no ip address
 bandwidth 38400
 encapsulation ppp
```

```
 dialer rotary-group 1
 ppp bap number default 5663457
!
interface bri 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
 ppp bap number default 5663458
!
interface dialer1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp multilink bap
 ppp bap call accept
 ppp bap link types isdn analog
 dialer load threshold 30
 ppp bap timeout pending 60
```

The following example configures a virtual template interface to use BACP in passive mode:

```
multilink virtual-template 1
!
interface virtual-template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp multilink bap
 ppp authentication chap callin
```

The bundle is created from any MMP-capable interface.

The following example creates a bundle on a BRI interface:

```
interface bri 0
 no ip address
 encapsulation ppp
 ppp multilink
 ppp bap number default 4000
 ppp bap number secondary 4001
```

Dialer Rotary Group with Different Dial-In Numbers

The following example configures a dialer rotary group that has four members, each with a different number, and that accepts incoming dial attempts. The dialer interface does not have a base phone number; the interface used to establish the first link in the multilink bundle will provide the appropriate number from its configuration.

```
interface bri 0
 no ip address
 encapsulation ppp
dialer rotary-group 1
 no fair-queue
 no cdp enable
 ppp bap number default 6666666
!
interface bri 1
 no ip address
 encapsulation ppp
dialer rotary-group 1
 no fair-queue
 no cdp enable
 ppp bap number default 6666667
!
```

```
interface bri 2
 no ip address
 encapsulation ppp
dialer rotary-group 1
 no fair-queue
 no cdp enable
 ppp bap number default 6666668
!
interface bri 3
 no ip address
 encapsulation ppp
dialer rotary-group 1
 no fair-queue
 no cdp enable
 ppp bap number default 6666669
!
interface dialer 1
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 300
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink bap
 ppp bap call accept
 ppp bap callback request
 ppp bap timeout pending 20
 ppp bap timeout response 2
 ppp bap max dial-attempts 2
 ppp bap monitor load
```

Passive Mode Dialer Rotary Group Members with One Dial-In Number

The following example, a dialer rotary group with two members each with the same number, accepts incoming dial attempts. The dialer interface has a base phone number because each of its member interfaces is in a hunt group and the same number can be used to access each individual interface.

```
interface bri 0
 no ip address
 encapsulation ppp
dialer rotary-group 1
 no fair-queue
 no cdp enable
!
interface bri 1
 no ip address
 encapsulation ppp
dialer rotary-group 1
 no fair-queue
 no cdp enable
!
interface dialer 1
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 300
 dialer-group 1
 no fair-queue
 no cdp enable
```

```
 ppp authentication chap
 ppp multilink bap
 ppp bap call accept
 ppp bap callback request
 ppp bap timeout pending 20
 ppp bap timeout response 2
 ppp bap max dial-attempts 2
 ppp bap monitor load
 ppp bap number default 6666666
```

PRI Interface with No Defined PPP BACP Number

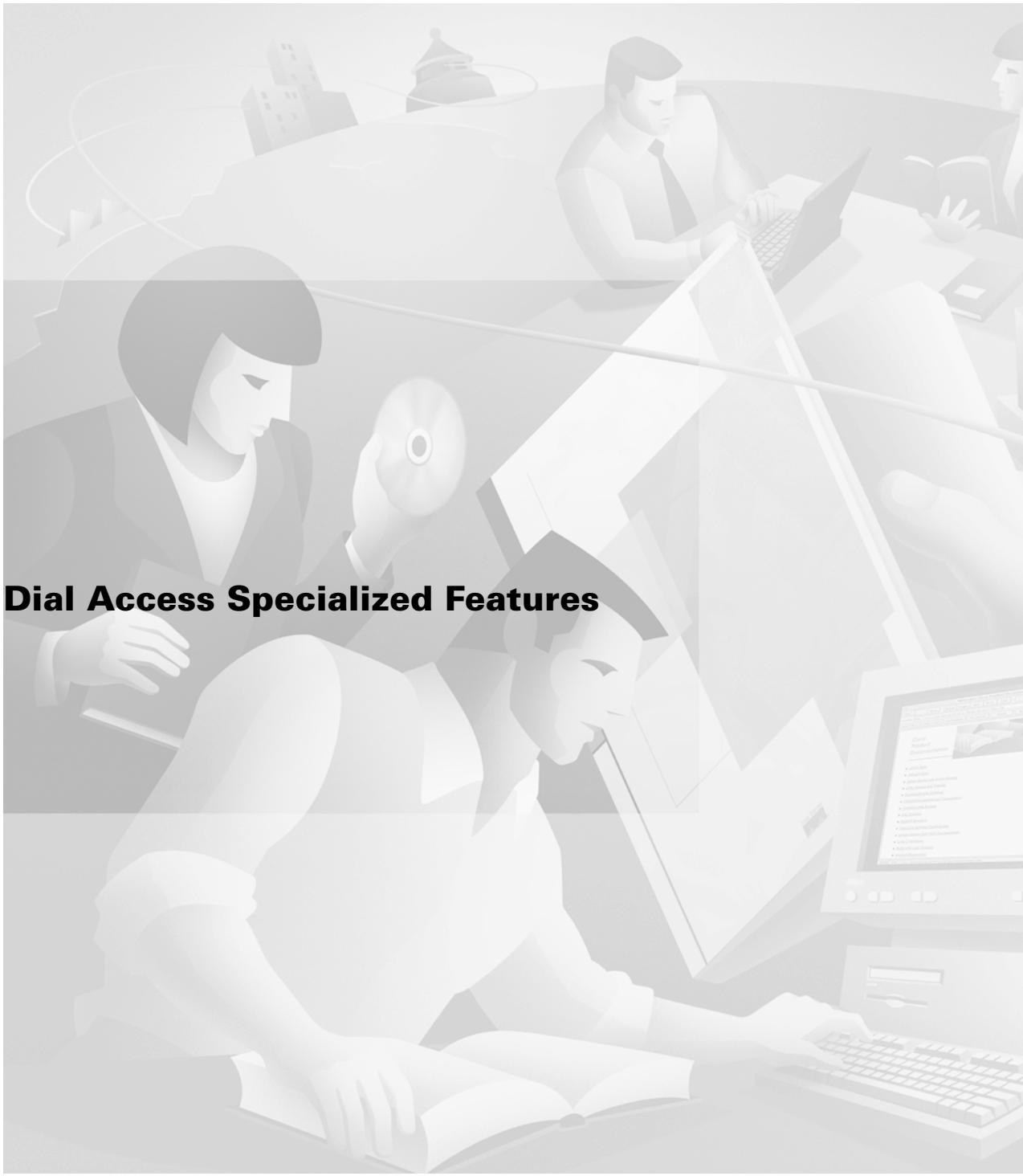
In the following example, a PRI interface has no BACP number defined and accepts incoming dial attempts (passive mode). The PRI interface has no base phone number defined, so each attempt to add a link would result in a delta of zero being provided to the calling peer. To establish the bundle, the peer should then dial the same number as it originally used.

```
interface serial 0:23
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 300
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink bap
 ppp bap call accept
 ppp bap callback request
 ppp bap timeout pending 20
 ppp bap timeout response 2
 ppp bap max dial-attempts 2
 ppp bap monitor load
```

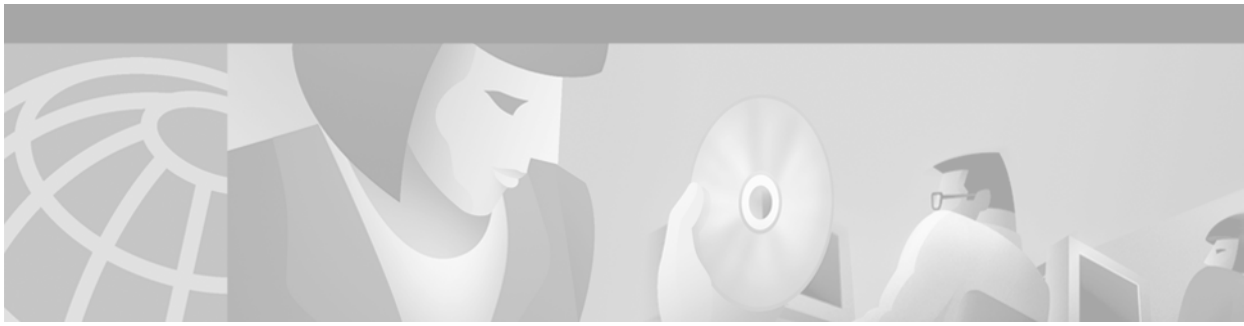
BRI Interface with No Defined BACP Number

In the following example, the BRI interface has no base phone number defined. The number that it uses to establish the bundle is that from the dialer map, and all phone delta operations are applied to that number.

```
interface bri 0
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 300
 dialer map ip 10.1.1.1 name bap_peer speed 56 19998884444
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink bap
 ppp bap call request
 ppp bap timeout pending 20
 ppp bap timeout response 2
 ppp bap max dial-attempts 2
 ppp bap monitor load
```



Dial Access Specialized Features



Configuring Large-Scale Dial-Out

This chapter describes how to configure large-scale dial-out. It includes the following main sections:

Consider these restrictions when configuring large-scale dial-out:

- Large-scale dial-out supports only IP over PPP encapsulation.
- Large-scale dial-out does not support tunneling protocols such as Layer 2 Forwarding Protocol (L2F) and Layer 2 Tunneling Protocol (L2TP).
- Virtual profiles depend on PPP authentication; however, this authentication can create a problem for Ascend devices, which do not allow devices to authenticate them when answering a call (bidirectional authentication is not supported).
- The IP address of the remote device must be known before dialing out. Large-scale dial-out does not support dynamic IP address assignment.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands mentioned in this chapter, refer to *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use *Cisco IOS Command Reference Master Index* or search online.

Large-Scale Dial-Out Overview

In previous dial-on-demand routing (DDR) networking strategies, only incoming calls could take advantage of features such as dialer and virtual profiles, Multichassis Multilink PPP (MMP) support, and the ability to use an authentication, authorization, and accounting (AAA) server to store attributes. MMP allows network access servers to be stacked together and appear as a single network access server chassis so that if one network access server fails, another network access server in the stack can accept calls. MMP also provides stacked network access servers access to a local Internet point of presence (POP) using a single telephone number. This capability allows for easy expansion and scalability and for assured fault tolerance and redundancy. Now, with large-scale dial-out, these features are available for both outgoing and incoming calls.

Large-scale dial-out eliminates the need to configure dialer maps on every network access server for every destination. Instead, you create remote site profiles that contain outgoing call attributes (telephone number, service type, and so on) on the AAA server. The profile is downloaded by the network access server when packet traffic requires a call to be placed to a remote site.

Additionally, large-scale dial-out addresses congestion management by seeking an uncongested, alternative network access server within the same POP when the designated primary network access server experiences port congestion.

Large-scale dial-out also enables scalable dial-out service to many remote sites across one or more Cisco network access servers or Cisco routers. This capability is especially beneficial to both Internet service providers (ISPs) and large-scale enterprise customers because it can simplify network configuration and management. Large-scale dial-out streamlines activities such as service maintenance and scheduled activities like application upgrades from a centralized location. Large enterprise networks such as those used by retail stores, supermarket chains, and franchise restaurants can use large-scale dial-out to easily update daily prices and inventory information from a central server to all branch locations in one process, using the same network access servers that they currently use for dial-in functions.

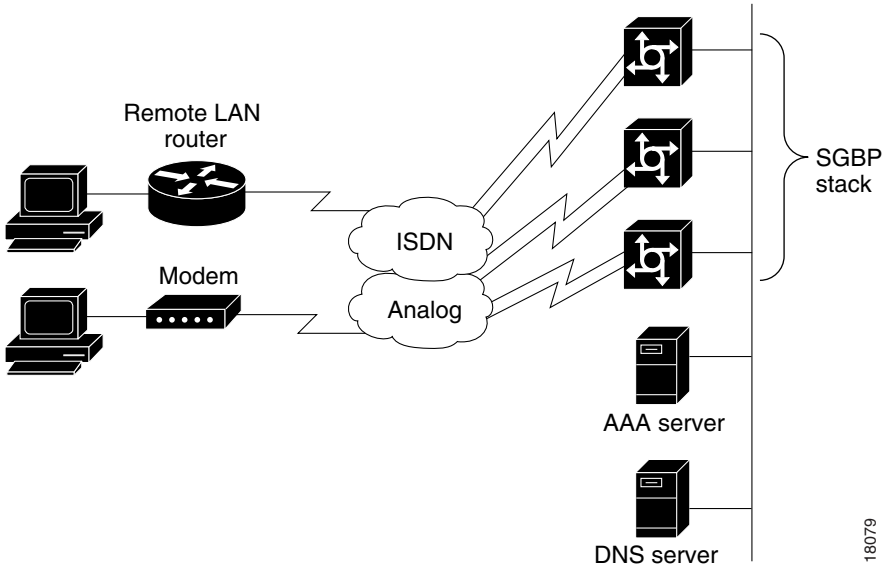
Additional benefits of using large-scale dial-out include the following:

- Allows dialing the same router from any router in a stack group. Using a primary network access server, you can configure static routes for a given remote host or network. If the primary network access server is congested or has no links available, it will search for an alternate server within the stack, and force that server to dial out.
- Eliminates the need to configure dialer maps in individual network access servers. The user profiles, along with dial parameters, can be centrally stored on an AAA server such as a Cisco Secure Access Control Server (ACS).
- Supports extended TACACS (also TACACS+), RADIUS using Cisco attribute-value (AV) pairs, and the Ascend proprietary RADIUS extension for dial-out operation.
- Provides a way to associate an IP address with a user name and user profile using the static route and host name association features. If there are no names on the IP static route, the Domain Name System (DNS) support function can be used to determine the user name that is associated with the IP address. If a name is not found, the destination IP address is used for the name.
- Allows dynamic static routes to be configured on the centralized AAA server, that is, static routes stored centrally on an AAA server that can be dynamically downloaded by the router as needed.
- Provides support for MMP and the Stack Group Bidding Protocol (SGBP). SGBP unites each Cisco access server in a virtual stack, which enables the access servers to become virtually tied together. If all ports on a given network access server are already being used, the other network access servers on the stack can be used for outbound calls. Single calls and multilink calls are now supported across the multichassis stack group.
- Supports dial-out over an asynchronous line, when a chat script is configured.
- Allows ports to be reserved for dial-in and dial-out.

Large-scale dial-out enables scalable dial-out service; that is, configuration information is stored in a central server, and many network access servers can access this information using either the RADIUS or extended TACACS protocols. One or more network access servers can advertise summary routes to the remote destinations and then dynamically download the dial-out profile configurations as needed.

Large-scale dial-out also allows dialing the same remote network or host from any router in a stack group. You configure static routes for a particular remote host or network on a router in a stack group that you designate as the primary network access server for that remote network or host. When a primary network access server experiences port congestion, it searches for an alternate network access server within the stack group to dial out and, when found, forces the alternate to dial the remote network. Figure 96 illustrates the large-scale dial-out solution.

Figure 96 Large-Scale Dial-Out Components



Large-scale dial-out relies on per-user static routes in AAA and redistributed static and redistributed connected routes to put better routes pointing to the same remote on the alternate network access server. You can use any routing protocol that supports redistributing static and connected routes and that supports Flash memory updates when a routing topology changes. The Open Shortest Path First (OSPF) and Enhanced Interior Gateway Routing Protocol (EIGRP) routing protocols are recommended.

Next Hop Definition

A next hop address or remote name that you define is used in an AAA server lookup to retrieve the user profile from the remote network or host. The name is passed to the AAA server by the router software.

Static Routes

Static routes can be dynamically downloaded from an AAA server by the network access servers or can be manually configured on the network access servers.

Dynamic static routes are installed on the network access server by an AAA server. The routes are downloaded at system startup and updated periodically, so that route changes are reflected within a configurable interval of time. Large-scale dial-out allows multiple AAA transactions with 50 static routes per AAA server transaction. There is no set limit for the number of AAA server transactions which can be configured, however configuring too many transactions may impact the performance of your network. Performance effects will depend on the configurations and platforms used in your network.

Stack Groups

The network access server stack group redistributes the routes of the remote networks. If the number is large, the routes are summarized. Packets destined for remote networks are routed to the primary network access server for the remote network.

If the static route that points to the next hop of the network access server has a name, that name with the -out suffix attached becomes the profile name. If no profile name is configured in the route statement that defines the remote location, the router can use reverse DNS lookup to map the IP route to a profile name. The next hop address on the static route is used in reverse DNS to obtain the name of the remote network. This name is then used in the AAA server lookup to retrieve the remote user profile. If no name is returned by DNS, the network access server uses the destination IP address with the -out suffix appended as the name.

If the primary network access server is congested, an alternate network access server may dial out. The primary network access server initiates stack group bidding for the outgoing call. The least congested network access server wins the bid and downloads the user profile. After a call is connected on an alternate network access server, a better per-user route from the AAA profile is installed on the alternate network access server. Subsequent packets destined for the remote network are routed to the alternate network access server while the call is connected. Packets stored in the dialer hold queue on the primary network access server are switched to the alternate network access server when the new route is distributed to the primary network access server.

How to Configure Large-Scale Dial-Out

To configure large-scale dial-out perform the tasks in the following sections:

- Complying with Large-Scale Dial-Out Prerequisites (Required)
- Establishing the Route to the Remote Network (As required)
- Enabling AAA and Static Route Download (Required)
- Enabling Access to the AAA Server (Required)
- Enabling Reverse DNS (Required)
- Enabling SGBP Dial-Out Connection Bidding (Required)
- Defining a User Profile (Required)

See the section "Monitoring and Maintaining the Large-Scale Dial-Out Network" later in this chapter for tips on maintaining large-scale dial-out. See the examples in the section "Configuration Examples for Large-Scale Dial-Out" at the end of this chapter for ideas on how you can implement large-scale dial-out in your network.

Complying with Large-Scale Dial-Out Prerequisites

The following prerequisites apply to large-scale dial-out:

- Virtual profiles depend on PPP authentication; therefore the network access server, the remote device, or both must authenticate the connection to use virtual profiles.
- You must configure SGBP to allow a primary network access server that is congested or otherwise unable to dial out to select an alternate network access server to dial out. Configure SGBP using the **sgbp group** and **sgbp member** global configuration commands before enabling the stack group to bid for dial-out connection. Configuring SGBP is described in the chapter "Configuring Multichassis Multilink PPP" in this publication. The *Cisco IOS Dial Technologies Command Reference* describes the commands to configure a stack group.

Additionally, all members of the stack group must be in the same routing autonomous system, and the **redistribute static** and **redistribute connected** commands must already be configured. The stack group supports all routing protocols, but routing protocols such as EIGRP and OSPF, which support redistributing static and connected routes and Flash memory updates when topology changes, are recommended.

- You must configure AAA network security services using the **aaa new-model**, **aaa authentication**, **aaa authorization**, and **aaa accounting** global configuration commands. For more information about AAA, see the chapter "AAA Overview" in the *Cisco IOS Security Configuration Guide*. The *Cisco IOS Security Command Reference* describes the commands to configure AAA.

  You will also need to configure your network access server to communicate with the applicable security server, either an extended TACACS or RADIUS daemon.

  If you are using RADIUS and Ascend attributes, use the **non-standard** keyword with the **radius-server host** command to enable your Cisco router, acting as a network access server, to recognize that the RADIUS security server is using a vendor-proprietary version of RADIUS. Use the **radius-server key** command to specify the shared secret text string used between your Cisco router and the RADIUS server. For more information, see the chapter "Configuring RADIUS" in the *Cisco IOS Security Configuration Guide*.

  If you are using extended TACACS, use the **tacacs-server host** command to specify the IP address of one or more extended TACACS daemons. Use the **tacacs-server key** command to specify the shared secret text string used between your Cisco router and the extended TACACS daemon. For more information, see the chapter about configuring extended TACACS in the *Cisco IOS Security Configuration Guide*.

Establishing the Route to the Remote Network

The task in this section is optional; you only need to perform it when routes will not be downloaded statically from the AAA server.

To establish a route to the remote network or host (next hop) that holds the user profile, use the **ip route** command in global configuration mode:

Command	Purpose
`Router(config)# ip route network-number [network-mask] {address | interface} [distance] [name name]`	Establishes a static route to a remote network to obtain a user profile.

The name you define is used in an AAA server lookup to retrieve the AAA profile of the remote network.

Enabling AAA and Static Route Download

AAA network security must be enabled before you perform the tasks in this section. For more information about enabling AAA, see the chapter "AAA Overview" in the *Cisco IOS Security Configuration Guide*.

Enabling the static route download feature allows static routes to be configured at a centrally located AAA server. Static routes are downloaded when the system is started, and you define a period of time between route updates when you enable the feature.

Note Static route download is not mandatory for the large-scale dial-out feature; however, it makes configuration of static routes more manageable by allowing the configuration to be centralized on a server.

To enable the static route download feature, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables the AAA server.
Step 2	`Router(config)# aaa route download` [*time*]	Downloads static routes from the AAA server periodically using the host name of the router.
Step 3	`Router(config)# aaa authorization configuration default` [**radius** | **tacacs+**]	Downloads configuration information from the AAA server.

Use the **show ip route** command to see the routes installed by these commands.

Enabling Access to the AAA Server

To configure the dialer interface to access the AAA server and retrieve the user profile, use the following command in interface configuration mode for a dialer rotary group leader:

Command	Purpose
`Router(config-if)# dialer aaa`	Allows the dialer to use the AAA server to locate profiles for dialing information.

Enabling Reverse DNS

To instruct the dialer to use reverse DNS on dial out, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# dialer dns`	Uses reverse DNS to obtain the name of the user profile of the remote network.

The user profile name passed to the AAA server by the system is *reverse-dns-name*-out; the -out suffix is automatically appended to the DNS name and is required to create unique dial-out and dial-in profiles.

Enabling SGBP Dial-Out Connection Bidding

You must configure SGBP before performing the tasks in this section. The chapter "Configuring Multichassis Multilink PPP" in this publication describes the tasks you perform to configure a stack group.

To configure stack group bidding, use the following command in global configuration mode:

Command	Purpose
`Router(config)# sgbp dial-bids`	Allows the stack group to bid for the dial-out call.

Once the stack group has been configured and enabled for dial-out connection bidding, configure the dialer interface to search for an alternate network access server in the event of port congestion. Use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# dialer congestion-threshold` *links*	Forces the dialer to search for another uncongested system in the stack group.
Step 2	`Router(config-if)# dialer reserved-links` {*dialin-link* | *dialout-link*}	Reserves links for dial in and dial-out.

See the section "Stack Group and Static Route Download Configuration Example" at the end of this chapter for an example of how to configure stack groups and static routes.

Defining a User Profile

Attributes are used to define specific AAA elements in a user profile. Large-scale dial-out supports a subset of Ascend AV pairs and RADIUS attributes, as well as a map class attribute that provides outbound dialing services, as described in Table 36.

The only required attribute is the Cisco AV pair outbound:dial-number; all others are optional. If the AAA server does not support Cisco AV pairs, attribute #227, Ascend-Dial-Number, can be substituted. If there are equivalent Cisco AV pairs and Ascend-specific attributes, Cisco recommends using the Cisco AV pairs.

For additional information about defining user profiles, see the chapter "RADIUS Attribute-Pairs" in the *CiscoSecure ACS for Windows NT User Guide 2.0* publication and the chapter "TACACS+ Attribute-Value Pairs" in the *Cisco IOS Security Configuration Guide*.

For an example of a user profile that uses the supported attributes, see the section "User Profile on an Ascend RADIUS Server for NAS1 Example" at the end of this chapter.

Note For the attributes listed in Table 4, the value of a string is 0 to 253 octects; the value of an integer is a 32-bit value ordered high byte first.

Table 36 Large-Scale Dial-Out Outbound Service Attributes

Number	Attribute	Description
Ascend AV Pairs		

#214	**Ascend-Send-Secret**	Specifies the password that the network access server uses when the remote site challenges the network access server to authenticate using either Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP).
		Cisco AV Pair:
		None
		TACACS+ Support:
		`service = outbound {`
		`  send-secret = VALUE`
		`}`
		Value:
		Password string
		Note The password is encrypted. This attribute requires a special RADIUS daemon that supports CHAP or PAP authentication.

#227	**Ascend-Dial-Number**	Defines the number to dial.
		Cisco AV Pair:
		`cisco-avpair="outbound:dial-number=VALUE"`
		TACACS+ Support:
		`service = outbound {`
		`  dial-number = VALUE`
		`}`
		Value:
		Dial string
		Note This attribute defines the plain dial number. It can be used in different profiles, whereas the callback-dialstring attribute is only for callbacks.

Table 36 Large-Scale Dial-Out Outbound Service Attributes (continued)

Number	Attribute	Description

#231	**Ascend-Send-Auth**	Specifies the authentication protocol that the network access server requests when initiating a connection using PPP. The answering side of the connection determines which authentication protocol, if any, that the connection uses. The network access server will refuse to negotiate PAP if CHAP is selected, but will negotiate CHAP if PAP is selected.
		Cisco AV Pair:
		`cisco-avpair="outbound:send-auth=VALUE"`
		TACACS+ Support:
		`service = outbound {`
		`  send-auth = none/pap/chap`
		`}`
		Value:
		0: Send-Auth-None
		1: Send-Auth-PAP
		2: Send-Auth-CHAP

#247	**Ascend-Data-SVC**	Specifies the type of data service that the link uses for outgoing calls.
		Cisco AV Pair:
		`cisco-avpair="outbound:data-service=VALUE"`
		TACACS+ Support:
		`service = outbound {`
		`  data-service = VALUE`
		`}`
		Value:
		0: Switched-Voice-Bearer

#248	**Ascend-Force-56**	Determines whether the network access server uses only the 56K portion of a channel, even when all 64K appear to be available.
		Cisco AV Pair:
		`cisco-avpair="outbound:force-56=VALUE"`
		TACACS+ Support:
		`service = outbound {`
		`  force-56 = VALUE`
		`}`
		Value:
		0: Force-56-No
		1: Force-56-Yes

Table 36 Large-Scale Dial-Out Outbound Service Attributes (continued)

Number	Attribute	Description
RADIUS (IETF) Attributes		

#10	**Framed-Routing**	Indicates a routing method when a router is used to access a network.
		Cisco AV Pair:
		None
		TACACS+ Support:
		`service = outbound {`
		`  routing = VALUE`
		`}`
		Value:
		0: None
		1: Broadcast
		2: Listen
		3: Broadcast-Listen
		Note This attribute is currently supported only for PPP service.

#19	**Callback-Number**	Defines a dialing string to be used for call back. (Service is both outbound and PPP.) **Cisco AV Pair:** `cisco-avpir="outbound:callback-dialstring=VALUE"` **TACACS+ Support:** Equivalent to the existing callback-dialstring attribute. **Value:** Dial string **Note** This is an alternate way of setting a callback number using a standard RADIUS attribute.

Table 36 Large-Scale Dial-Out Outbound Service Attributes (continued)

Number	Attribute	Description
#61	**NAS-Port-Type**	Indicates the type of physical port that the network access server is using to authenticate the user. **Cisco AV Pair:** None **TACACS+ Support:** None **Value:** 0: Asynchronous 1: Synchronous 2: ISDN-Synchronous **Note** This attribute is currently supported only for PPP service.
Map Class Attribute		

(unnumbered)	**map-class**	Allows the user profile to reference information configured in a map class of the same name on the network access server that dials out.
		Cisco AV Pair:
		`cisco-avpair="outbound:map-class=VALUE"`
		TACACS+ Support:
		`service = outbound {`
		`  map-class = VALUE`
		`}`
		Value:
		Name string, which must match the name of a map class on the dial-out network access server.

Monitoring and Maintaining the Large-Scale Dial-Out Network

To monitor and maintain a large-scale dial-out network, use any of the following commands in EXEC mode:

Command	Purpose
`Router> clear dialer sessions`	Removes all dialer sessions and disconnects links.
`Router> clear ip route download {* | network-number network-mask | reload}`	Removes all or specified IP routes on the router. With the **reload** option, forces reload of dynamic static routes before the update timer expires.
`Router> show dialer sessions`	Displays all dialer sessions.
`Router> show ip route [static [download]]`	Displays all static IP routes or those installed using the AAA route download function.

Configuration Examples for Large-Scale Dial-Out

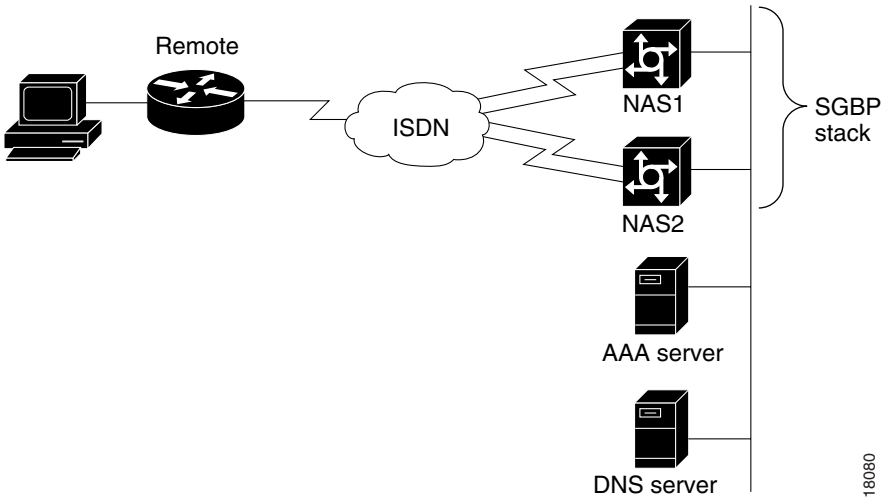
The following sections provide examples of how you can configure large-scale dial-out in your network:

- Stack Group and Static Route Download Configuration Example
- User Profile on an Ascend RADIUS Server for NAS1 Example
- Asynchronous Dialing Configuration Examples

Stack Group and Static Route Download Configuration Example

The following example configures NAS1 as the primary network access server and NAS2 as the secondary network access server, in a stack group for dial-out. The remote router is configured to answer calls. Figure 97 illustrates the configuration.

Figure 97 Stack Group and Static Route Download Configuration



At the console for NAS1, ping 20.1.1.1. This action creates a multilink bundle with two links. NAS1 dials out the first link, and NAS2 dials out the second link. The router named Remote is using the CHAP host name echo-8.cisco.com.

A user profile for NAS1 on an Ascend RADIUS server is listed in the section "User Profile on an Ascend RADIUS Server for NAS1 Example" later in this chapter.

Primary Network Access Server Configuration for NAS1

```
version 12.0
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
!
hostname NAS1
!
aaa new-model
aaa authentication ppp default radius local
aaa authorization network default radius none
aaa authorization configuration default radius
aaa route download 720
enable password 7 1236173C1B0F
!
username NAS2 password 7 05080F1C2243
username NAS1 password 7 030752180500
username dialbid password 7 121A0C041104
username echo-8.cisco.com password 7 02050D480809
ip subnet-zero
ip domain-name cisco.com
ip name-server 172.31.2.132
ip name-server 172.22.30.32
!
virtual-profile virtual-template 2
!
sgbp group dialbid
sgbp seed-bid offload
sgbp member NAS2 172.21.17.17
sgbp dial-bids
isdn switch-type basic-5ess
!
!
interface Ethernet 0
 ip address 172.21.17.18 255.255.255.0
 no ip directed-broadcast
 no ip mroute-cache
 media-type 10BaseT
 no cdp enable
!
interface Virtual-Template 1
 ip address 10.1.1.1 255.255.255.252
 no ip directed-broadcast
!
interface Virtual-Template 2
 ip unnumbered Virtual-Template 1
 no ip directed-broadcast
 ppp multilink
 multilink load-threshold 1 outbound
!
interface BRI 0
 description PBX 60043
 no ip address
 no ip directed-broadcast
 encapsulation ppp
```

```
 dialer rotary-group 1
 isdn switch-type basic-5ess
 no fair-queue
!
interface Dialer 1
 ip unnumbered Ethernet 0
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 dialer in-band
 dialer dns
 dialer aaa
 dialer hold-queue 5
 dialer congestion-threshold 5
 dialer reserved-links 1 0
 dialer-group 1
 no fair-queue
 ppp authentication chap callin
 ppp multilink
!
router eigrp 200
 redistribute connected
 redistribute static
 network 172.21.0.0
!
ip default-gateway 172.21.17.1
ip classless
ip route 0.0.0.0 0.0.0.0 172.21.17.1
!
dialer-list 1 protocol ip permit
radius-server host 172.31.61.87 auth-port 1645 acct-port 1646
radius-server key foobar
!
end
```

Secondary Network Access Server Configuration for NAS2

```
version 12.0
service timestamps debug datetime msec
service timestamps log uptime
service password-encryption
!
hostname NAS2
!
boot system flash
aaa new-model
aaa authentication ppp default radius local
aaa authorization network default radius none
aaa authorization configuration default radius
enable password 7 022916700202
!
username NAS1 password 7 104D000A0618
username dialbid password 7 070C285F4D06
username echo-8.cisco.com password 7 0822455D0A16
ip subnet-zero
ip domain-name cisco.com
ip name-server 172.22.30.32
ip name-server 172.31.2.132
!
virtual-profile virtual-template 2
!
sgbp group dialbid
sgbp member NAS1 172.21.17.18
```

```
sgbp dial-bids
isdn switch-type basic-5ess
!
interface Ethernet 0
 ip address 172.21.17.17 255.255.255.0
 no ip directed-broadcast
 media-type 10BaseT
!
interface Virtual-Template 1
 ip address 10.1.1.1 255.255.255.252
 no ip directed-broadcast
!
interface Virtual-Template 2
 ip unnumbered Virtual-Template 1
 no ip directed-broadcast
 ppp multilink
 multilink load-threshold 1 outbound
!
interface BRI 0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer rotary-group 0
 isdn switch-type basic-5ess
 no fair-queue
!
interface Dialer 0
 ip unnumbered Ethernet 0
 no ip directed-broadcast
 encapsulation ppp
 dialer in-band
 dialer dns
 dialer aaa
dialer hold-queue 5
 dialer congestion-threshold 5
 dialer reserved-links 1 0
 dialer-group 1
 no fair-queue
 ppp authentication chap callin
 ppp multilink
!
router eigrp 200
 redistribute connected
 redistribute static
 network 172.21.0.0
!
ip default-gateway 172.21.17.1
ip classless
ip route 0.0.0.0 0.0.0.0 172.21.17.1
!
dialer-list 1 protocol ip permit
!
radius-server host 172.31.61.87 auth-port 1645 acct-port 1646
radius-server key foobar
!
end
```

Remote Router Configuration

```
version 12.0
service timestamps debug datetime msec
service timestamps log uptime
service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname Remote
!
boot system flash
enable password 7 002B012D0D5F
!
username dialbid password 7 14141B180F0B
ip subnet-zero
no ip domain-lookup
!
isdn switch-type basic-5ess
!
interface Loopback 0
 ip address 172.31.229.41 255.255.255.255
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback 1
 ip address 10.1.1.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback 2
 ip address 10.1.2.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Loopback 3
 ip address 10.3.1.1 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet 0
 ip address 172.21.12.15 255.255.255.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
!
interface BRI 0
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer rotary-group 3
 dialer-group 1
 isdn switch-type basic-5ess
 no fair-queue
!
interface Dialer 3
 ip unnumbered Loopback 0
 no ip directed-broadcast
```

```
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 dialer in-band
 dialer idle-timeout 10000
 dialer-group 1
 no fair-queue
 ppp authentication chap callin
 ppp chap hostname echo-8.cisco.com
 ppp chap password 7 045802150C2E
 ppp multilink
!
ip default-gateway 172.21.12.1
ip classless
ip route 0.0.0.0 0.0.0.0 1.1.1.1
!
dialer-list 1 protocol ip permit
```

User Profile on an Ascend RADIUS Server for NAS1 Example

The following example shows a dial-out profile and a static route download profile in AAA. The dial-out profile username must have "-out" appended to it. The static route download profile username always has "-*N*" appended. The router downloads NAS1-1, NAS1-2, through NAS1-*N*. When NAS1-*N* fails, the router does not try NAS1-*N*+1. The static route download profile cannot have more than 50 static routes defined.

```
echo-8.cisco.com-out  Password = "cisco", User-Service-Type = Outbound-User
     cisco-avpair = "outbound:addr=172.31.229.41",
     cisco-avpair = "outbound:dial-number=60039",

NAS1-1 Password = "cisco" User-Service-Type = Outbound-User,
     cisco-avpair = "ip:route=10.1.3.0 255.255.255.0 172.31.229.41 200",
     cisco-avpair = "ip:route=10.1.2.0 255.255.255.0 172.31.229.41 200",
     cisco-avpair = "ip:route=10.1.1.0 255.255.255.0 172.31.229.41 200",
     cisco-avpair = "ip:route=172.31.229.41 255.255.255.255 Dialer1 200 name
echo-8.cisco.com"
```

Note Note that all text between quotation marks must be typed on one line.

Static routes can also be defined using the Framed-Route Internet Engineering Task Force (IETF) standard. The following example shows how the previous example for NAS1 would look using the Framed-Route IETF standard:

```
NAS1-1 Password = "cisco" User-Service-Type = Outbound-User,
Framed-Route = "10.1.3.0/24 172.31.229.41.200",
Framed-Route = "10.1.2.0/24 172.31.229.41.200",
Framed-Route = "10.1.1.0/24 172.31.229.41.200",
Framed-Route = "172.31.229.41/32 Dialer1 200 name echo-8.cisco.com"
```

Asynchronous Dialing Configuration Examples

Large-scale dial-out supports dialing out using an asynchronous line. This type of dialing requires that a chat script be configured and that the **script dialer** command be configured in the line commands for any asynchronous interface that may be dialing out. The following examples are provided in this section:

- Asynchronous Dialing Example
- Asynchronous and Synchronous Dialing Example

Asynchronous Dialing Example

The following example shows an asynchronous dialing configuration:

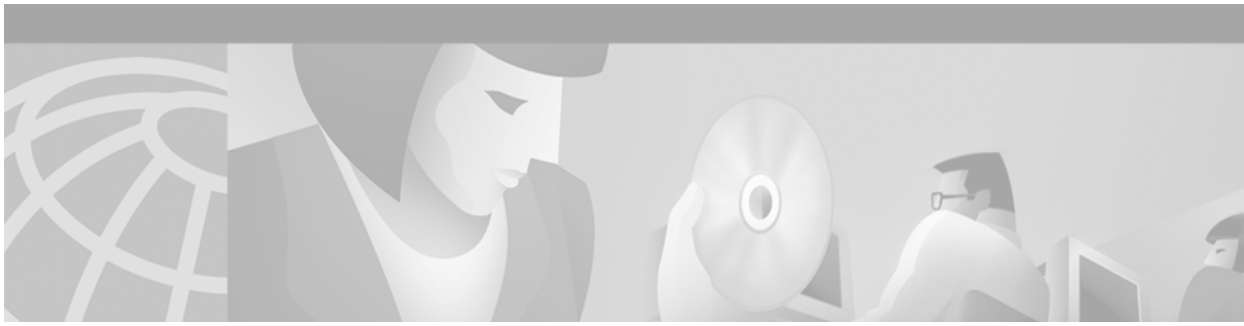
```
chat-script dial "" "ATZ" OK "ATDT\T" TIMEOUT 60 CONNECT
!
interface Async 1
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer in-band
 dialer rotary-group 0
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
!
interface Dialer 0
 ip address 172.21.30.32 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 bandwidth 64
 dialer in-band
 dialer idle-timeout 60
 dialer enable-timeout 10
 dialer hold-queue 50
 dialer-group 1
 no cdp enable
!
line 1
 script dialer dial
 modem InOut
 transport input all
```

Asynchronous and Synchronous Dialing Example

The following example creates a dialer rotary group for the asynchronous interfaces and a dialer rotary group for the PRI interfaces. Any dial-in or dial-out reservations are applied only to the PRI dialer interface. In the following configuration example:

- Destinations that require modem calls have static routes that point to Dialer 0.
- Destinations that require digital connections have static routes that point to Dialer 1.
- The **dialer reserved-links** command applies to all connections made over the PRI interfaces in dialer rotary group 1, even if they come from an asynchronous interface.

```
chat-script dial "" "ATZ" OK "ATDT\T" TIMEOUT 60 CONNECT
!
interface Serial 0:23
 no ip address
 no ip directed-broadcast
 no keepalive
 dialer rotary-group 1
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 no cdp enable
!
interface Async 1
 no ip address
 no ip directed-broadcast
 encapsulation ppp
 dialer in-band
 dialer rotary-group 0
 async dynamic address
 async dynamic routing
 async mode dedicated
 no cdp enable
!
interface Dialer 0
 ip address 172.21.30.32 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 no ip mroute-cache
 bandwidth 64
 dialer in-band
 dialer dns
 dialer aaa
 dialer idle-timeout 60
 dialer enable-timeout 10
 dialer hold-queue 50
 dialer-group 1
 no cdp enable
!
interface Dialer 1
 ip address unnumbered eth0
 no ip directed-broadcast
 dialer in-band
 dialer dns
 dialer aaa
 dialer reserved-links 22 0
 no cdp enable
!
line 1
 script dialer dial
 modem InOut
 transport input all
```



Configuring per-User Configuration

This chapter describes per-user configuration, a large-scale dial solution. It includes the following main sections:

This set of features is supported on all platforms that support Multilink PPP (MLP).

A virtual access interface created dynamically for any user dial-in session is deleted when the session ends. The resources used during the session are returned for other dial-in uses.

When a specific user dials in to a router, the use of a per-user configuration from an authentication, authorization, and accounting (AAA) server requires that AAA is configured on the router and that a configuration for that user exists on the AAA server.

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2 and the *Cisco IOS Security Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Per-User Configuration Overview

Per-user configuration provides a flexible, scalable, easily maintained solution for customers with a large number of dial-in users. This solution can tie together the following dial-in features:

- Virtual template interfaces, generic interface configuration and router-specific configuration information stored in the form of a virtual template interface that can be applied (*cloned*) to a virtual access interface each time any user dials in. This configuration is described in the chapter "Configuring Virtual Template Interfaces" in this publication.
- AAA per-user security and interface configuration information stored on a separate AAA server and sent by the AAA server to the access server or router in response to authorization requests during the PPP authentication phase. The per-user configuration information can add to or override the generic configuration on a virtual interface.

- Virtual profiles, which can use either or both of the two sources of information listed in the previous bullets for virtual interface configuration. When a user dials in, virtual profiles can apply the generic interface configuration and then apply the per-user configuration to create a unique virtual access interface for that user. This configuration is described in the chapter "Configuring Virtual Profiles" in this publication.

The per-user configuration feature provides these benefits:

- Maintenance ease for service providers with a large number of access servers and a very large number of dial-in users. Service providers need not update all their routers and access servers when user-specific information changes; instead, they can update one AAA server.
- Scalability. By separating generic virtual interface configuration on the router from the configuration for each individual, Internet service providers and other enterprises with large numbers of dial-in users can provide a uniquely configured interface for each individual user. In addition, by separating the generic virtual interface configuration from the physical interfaces on the router, the number and types of physical interfaces on the router or access server are not intrinsic barriers to growth.

General Operational Processes

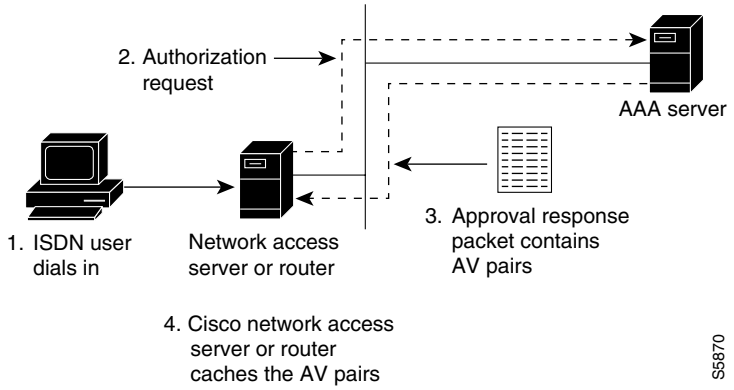
In general, the per-user configuration process on the Cisco router or network access server proceeds as follows:

1. The user dials in.
2. The authentication and authorization phases occur.
   a. If AAA is configured, the router sends an authorization request to the AAA server.
   b. If the AAA server has information (attribute-value or AV pairs, or other configuration parameters) that defines a configuration for the specific user, the server includes it in the information in the approval response packet.

      Figure 98 illustrates the request and response part of the process that happens when a user dials in, given that AAA is configured and that the AAA server has per-user configuration information for the dial-in user.
   c. The router looks for AV pairs in the AAA approval response.
   d. The router caches the configuration parameters.

Note TACACS servers treat authentication and authorization as two phases; RADIUS servers combine authentication and authorization into a single step. For more detailed information, refer to your server documentation.

Figure 98 Per-User Configuration Authentication and Authorization



3. A virtual access interface is created for this user.
   a. The router finds the virtual template that is set up for virtual profiles, if any, and applies the commands to the virtual access interface.
   b. The router looks for the AV pairs to apply to this virtual access interface to configure it for the dial-in user.
   c. The AV pairs are sent to the Cisco IOS command-line parser, which interprets them as configuration commands and applies them to configure this virtual access interface.

The result of this process is a virtual access interface configured uniquely for the dial-in user.

When the user ends the call, the virtual access interface is deleted and its resources are returned for other dial-in uses.

Note The use of virtual profiles can modify the process that occurs between the user dial-in and the use of AAA configuration information. For more information, see the chapter "Configuring Virtual Profiles" in this publication.

Operational Processes with IP Address Pooling

During IP Control Protocol (IPCP) address negotiation, if an IP pool name is specified for a user, the network access server checks whether the named pool is defined locally. If it is, no special action is required and the pool is consulted for an IP address.

If the required pool is not present (either in the local configuration or as a result of a previous download operation), an authorization call to obtain it is made using the special username:

```
pools-nas-name
```

where *nas-name* is the configured name of the network access server. In response, the AAA server downloads the configuration of the required pool.

This pool username can be changed using Cisco IOS configuration, for example:

```
aaa configuration config-name nas1-pools-definition.cisco.us
```

This command has the effect of changing the username that is used to download the pool definitions from the default name "pools-*nas-name*" to "nas1-pools-definition.cisco.com."

On a TACACS+ server, the entries for an IP address pool and a user of the pool might be as follows:

```
user = nas1-pools {
     service = ppp protocol = ip {
        pool-def#1 = "aaa 10.0.0.1 10.0.0.3"
        pool-def#2 = "bbb 10.1.0.1 10.1.0.10"
        pool-def#3 = "ccc 10.2.0.1 10.2.0.20"
        pool-timeout=60
    }
}

user = georgia {
    login = cleartext lab
    service = ppp protocol = ip {
                  addr-pool=bbb
    }
}
```

On a RADIUS server, the entries for the same IP address pool and user would be as follows:

```
nas1-pools      Password = "cisco" User-Service-Type=Outbound-User
        cisco-avpair = "ip:pool-def#1=aaa 10.0.0.1 10.0.0.3",
        cisco-avpair = "ip:pool-def#2=bbb 10.1.0.1 10.1.0.10",
        cisco-avpair = "ip:pool-def#3=ccc 10.2.0.1 10.2.0.20",
        cisco-avpair = "ip:pool-timeout=60"

georgia Password = "lab"
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
                cisco-avpair = "ip:addr-pool=bbb"
```

Note This entry specifies a User-Service-Type of Outbound-User. This attribute is supplied by the network access server to prevent ordinary logins from using the well-known username and password combination of nas1-pools/cisco.

Pools downloaded to a Cisco network access server are not retained in nonvolatile memory and automatically disappear whenever the access server or router restarts. Downloaded pools can also be made to time out automatically by adding a suitable AV pair. For more information, see the section "Supported Attrubutes for AV Pairs" and the pool-timeout attribute in Table 37. Downloaded pools are marked as *dynamic* in the output of the **show ip local pool** command.

Deleting Downloaded Pools

To delete downloaded pools, you can do either of the following:

- Manually delete the definition from the network access server. For example, if "bbb" is the name of a downloaded pool, you can enter the Cisco IOS **no ip local pool bbb** command.

  Deleting a pool definition does not interrupt service for current users. If a pool is deleted and then redefined to include a pool address that is currently allocated, the new pool understands and tracks the address as expected.

- Set an AV pair pool-timeout value; this is a more desirable solution.

  The pool-timeout AV pair starts a timer when the pool is downloaded. Once the timer expires, the pools are deleted. The next reference to the pools again causes an authorization call to be made, and the pool definition is downloaded again. This method allows definitions to be made and changed on the AAA server and propagated to network access servers.

Supported Attributes for AV Pairs

Table 37 provides a partial list of the Cisco-specific supported attributes for AV pairs that can be used for per-user virtual interface configuration. For complete lists of Cisco-specific, vendor-specific, and TACACS+ supported attributes, see the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference*.

Table 37 Partial List of Cisco-Specific Supported AV Pair Attributes

Attribute	Meaning
inacl#	An input access list definition. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For Internet Protocol Exchange (IPX), only extended syntax is recognized. The value of this attribute is the text that comprises the body of a named access list definition.
outacl#[1]	An output access list definition. For IP, standard or extended access list syntax can be used. For IPX, only extended syntax is recognized. The value of this attribute is the text that comprises the body of a named access list definition.
rte-fltr-in#	An input route filter. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For IPX, only extended syntax is recognized. The first line of this filter must specify a routing process. Subsequent lines comprise the body of a named access list.
rte-fltr-out#	An output route filter. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For IPX, only extended syntax is recognized. The first line of this filter must specify a routing process. Subsequent lines comprise the body of a named access list.
route#[2]	Static routes, for IP and IPX. The value is text of the form *destination-address mask* [*gateway*].
sap#	IPX static Service Advertising Protocol (SAP). The value is text from the body of an **ipx sap** configuration command.
sap-fltr-in#	IPX input SAP filter. Only extended access list syntax is recognized. The value is text from the body of an extended IPX **access-list** configuration command. (The Novell socket number for SAP filtering is 452.)
sap-fltr-out#	IPX output SAP filter. Only extended **access-list** command syntax is recognized. The value is text from the body of an extended IPX access-list configuration command.
pool-def#	An IP pool definition. The value is text from the body of an **ip local pool** configuration command.
pool-timeout	An IP pool definition. The body is an integer representing a timeout, in minutes.

1. The “outacl” attribute still exists and retains its old meaning.
2. The “route” attribute, without a trailing #, is still recognized for backward compatibility with the TACACS+ protocol specification, but if multiple static routes are required in TACACS+, full “route#” names will need to be employed.

Table 38 provides examples for each attribute on an AAA TACACS+ server.

Table 38 TACACS+ Server AV Pair Examples for Each Attribute

Attribute	TACACS+ Server Examples

inacl#	**IP:**
	`inacl#3="permit ip any any precedence immediate"`
	`inacl#4="deny igrp 10.0.1.2 255.255.0.0 any"`
	IPX:
	`inacl#1="deny 3C01.0000.0000.0001"`
	`inacl#2="deny 4C01.0000.0000.0002"`

outacl#	`outacl#2="permit ip any any precedence immediate"` `outacl#3="deny igrp 10.0.9.10 255.255.0.0 any"`

rte-fltr-in#	**IP:**
	`rte-fltr-in#1="router igrp 60"`
	`rte-fltr-in#3="permit 10.0.3.4 255.255.0.0"`
	`rte-fltr-in#4="deny any"`
	IPX:
	`rte-fltr-in#1="deny 3C01.0000.0000.0001"`
	`rte-fltr-in#2="deny 4C01.0000.0000.0002"`

rte-fltr-out#	`rte-fltr-out#1="router igrp 60"`
	`rte-fltr-out#3="permit 10.0.5.6 255.255.0.0"`
	`rte-fltr-out#4="permit any"`

route#	**IP:**
	`route#1="10.0.0.0 255.0.0.0 1.2.3.4"`
	`route#2="10.1.0.0 255.0.0.0"`
	IPX:
	`route#1="4C000000 ff000000 10.12.3.4"`
	`route#2="5C000000 ff000000 10.12.3.5"`

sap#	`sap#1="4 CE1-LAB 1234.0000.0000.0001 451 4"` `sap#2="5 CE3-LAB 2345.0000.0000.0001 452 5"`
sap-fltr-in#	`sap-fltr-in#1="deny 6C01.0000.0000.0001"` `sap-fltr-in#2="permit -1"`
sap-fltr-out#	`sap-fltr-out#1="deny 6C01.0000.0000.0001"` `sap-fltr-out#2="permit -1"`

pool-def#	`pool-def#1 = "aaa 10.0.0.1 1.0.0.3"`
	`pool-def#2 = "bbb 10.1.0.1 2.0.0.10"`
	`pool-def#3 = "ccc 10.2.0.1 3.0.0.20"`

pool-timeout	`pool-timeout=60`

Table 39 provides examples for each attribute on an AAA RADIUS server.

Table 39 RADIUS Server AV Pair Examples for Each Attribute

Attribute	RADIUS Server Examples
lcp:interface-config[1]	`cisco-avpair = "lcp:interface-config=ip address 10.0.0.0 255.255.255.0",`
inacl#	`cisco-avpair = "ip:inacl#3=permit ip any any precedence immediate",` `cisco-avpair = "ip:inacl#4=deny igrp 10.0.1.2 255.255.0.0 any",`

Table 39 RADIUS Server AV Pair Examples for Each Attribute (continued)

Attribute	RADIUS Server Examples
outacl#	`cisco-avpair = "ip:outacl#2=permit ip any any precedence immediate",` `cisco-avpair = "ip:outacl#3=deny igrp 10.0.9.10 255.255.0.0 any",`

rte-fltr-in#	**IP:**
	`cisco-avpair = "ip:rte-fltr-in#1=router igrp 60",`
	`cisco-avpair = "ip:rte-fltr-in#3=permit 10.0.3.4 255.255.0.0",`
	`cisco-avpair = "ip:rte-fltr-in#4=deny any",`
	IPX:
	`cisco-avpair = "ipx:rte-fltr-in=deny 3C01.0000.0000.0001",`

rte-fltr-out#	`cisco-avpair = "ip:rte-fltr-out#1=router igrp 60",`
	`cisco-avpair = "ip:rte-fltr-out#3=permit 10.0.5.6 255.255.0.0",`
	`cisco-avpair = "ip:rte-fltr-out#4=permit any",`

route#	**IP:**
	`cisco-avpair = "ip:route=3.10.0.0 255.0.0.0 1.2.3.4",`
	`cisco-avpair = "ip:route=4.10.0.0 255.0.0.0",`
	IPX:
	`cisco-avpair = "ipx:route=4C000000 ff000000 10.12.3.4",`
	`cisco-avpair = "ipx:route=5C000000 ff000000 10.12.3.5"`

sap#	`cisco-avpair = "ipx:sap=4 CE1-LAB 1234.0000.0000.0001 451 4",` `cisco-avpair = "ipx:sap=5 CE3-LAB 2345.0000.0000.0001 452 5",`
sap-fltr-in#	`cisco-avpair = "ipx:sap-fltr-in=deny 6C01.0000.0000.0001",` `cisco-avpair = "ipx:sap-fltr-in=permit -1"`
sap-fltr-out#	`cisco-avpair = "ipx:sap-fltr-out=deny 6C01.0000.0000.0001",` `cisco-avpair = "ipx:sap-fltr-out=permit -1"`

pool-def#	`cisco-avpair = "ip:pool-def#1=aaa 10.0.0.1 1.0.0.3",`
	`cisco-avpair = "ip:pool-def#2=bbb 10.1.0.1 2.0.0.10",`
	`cisco-avpair = "ip:pool-def#3=ccc 10.2.0.1 3.0.0.20",`

pool-timeout	`cisco-avpair = "ip:pool-timeout=60"`

1. This attribute is specific to RADIUS servers. It can be used to add Cisco IOS interface configuration commands to specific user configuration information.

How to Configure a AAA Server for Per-User Configuration

The configuration requirements and the structure of per-user configuration information is set by the specifications of each type of AAA server. Refer to your server documentation for more detailed information. The following sections about TACACS and RADIUS servers are specific to per-user configuration:

- Configuring a Freeware TACACS Server for Per-User Configuration (As required)
- Configuring a CiscoSecure TACACS Server for Per-User Configuration (As required)
- Configuring a RADIUS Server for Per-User Configuration (As required)

See the section "Monitoring and Debugging Per-User Configuration Settings" later in this chapter for tips on troubleshooting per-user configuration settings. See the section "Configuration Examples for Per-User Configuration" at the end of this chapter for examples of configuring RADIUS and TACACS servers.

Configuring a Freeware TACACS Server for Per-User Configuration

On a TACACS server, the entry in the user file takes a standard form. In the freeware version of TACACS+, the following lines appear in order:

- "User =" followed by the username, a space, and an open brace
- Authentication parameters
- Authorization parameters
- One or more AV pairs
- End brace on a line by itself

The general form of a freeware TACACS user entry is shown in the following example:

```
user = username {
     authentication parameters go here
     authorization parameters go here
}
```

The freeware TACACS user entry form is also shown by the following examples for specific users:

```
user= Router1
  Password= cleartext welcome
  Service= PPP protocol= ip {
        ip:route=10.0.0.0 255.0.0.0
        ip:route=10.1.0.0 255.0.0.0
        ip:route=10.2.0.0 255.0.0.0
        ip:inacl#5=deny 10.5.0.1
}

user= Router2
  Password= cleartext lab
  Service= PPP protocol= ip {
        ip:addr-pool=bbb
}
```

For more requirements and detailed information, refer to your AAA server documentation.

Configuring a CiscoSecure TACACS Server for Per-User Configuration

The format of an entry in the user file in the AAA database is generally name = value. Some values allow additional subparameters to be specified and, in these cases, the subparameters are enclosed in braces ({}). The following simple example depicts an AAA database showing the default user, one group, two users that belong to the group, and one user that does not:

```
# Sample AA Database 1
unknown_user = {
  password = system #Use the system's password file (/etc/passwd)
}
group = staff {
  # Password for staff who do not have their own.
  password = des "sefjkAlM7zybE"
  service = shell {
    # Allow any commands with any attributes.
    default cmd = permit
    default attribute = permit
  }
```

```
}
user = joe {  # joe uses the group password.

  member = "staff"
}
user = pete {  # pete has his own password.
  member = "staff"
  password = des "alkd9Ujiqp2y"
}
user = anita {
  # Use the "default" user password mechanism defined above.
  service = shell {
    cmd = telnet {  # Allow Telnet to any destination
    }
  }
}
```

For more information about the requirements and details of configuring the CiscoSecure server, see the *CiscoSecure UNIX Server User Guide*.

Configuring a RADIUS Server for Per-User Configuration

On a RADIUS server, the format of an entry in the users file includes the following lines in order:

- Username and password
- User service type
- Framed protocol
- One or more AV pairs

Note All these AV pairs are vendor specific. To use them, RADIUS servers must support the use of vendor-specific AV pairs. Patches for some servers are available from the Cisco Consulting Engineering (CE) customer-support organization.

The structure of an AV pair for Cisco platforms starts with *cisco-avpair* followed by a space, an equal sign, and another space. The rest of the line is within double quotation marks and, for all lines but the last, ends with a comma. Inside the double quotation marks is a phrase indicating the supported attribute, another equal sign, and a Cisco IOS command. The following examples show two different partial user configurations on a RADIUS server.

Router1

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.1.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.2.0.0 255.0.0.0",
cisco-avpair = "ip:inacl#5=deny 10.5.0.1"
```

Router2

```
Password = "lab"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:addr-pool=bbb"
```

Monitoring and Debugging Per-User Configuration Settings

Per-user configuration information exists on AAA servers only and is configured there, as described in the "How to Configure a AAA Server for Per-User Configuration" section.

For more information about configuring an application that can tie AAA per-user configuration information to generic interface and router configuration, see the chapter "Configuring Virtual Profiles" in this publication. Virtual profiles are required for combining per-user configuration information and generic interface and router configuration information to create virtual access interfaces for individual ISDN B channels.

However, you can monitor and debug the per-user configuration settings on the router or access server that are set from an AAA server. Table 40 indicates some of the commands to use for each attribute.

Table 40 Monitoring and Debugging Per-User Configuration Commands

Attribute	show Commands	debug Commands
inacl# outacl#	**show ip access-list** **show ip interface** *interface* **show ipx access-list** **show ipx interface**	**debug aaa authorization** **debug aaa per-user**
rte-fltr-in# rte-fltr-out#	**show ip access-list** **show ip protocols**	**debug aaa authorization** **debug aaa per-user**
route#	**show ip route** **show ipx route**	**debug aaa authorization** **debug aaa per-user**
sap#	**show ipx servers**	**debug aaa authorization** **debug aaa per-user**
sap-fltr-in# sap-fltr-out#	**show ipx access-list** **show ipx interface**	**debug aaa authorization** **debug aaa per-user**
pool-def# pool-timeout	**show ip local pool** [*name*]	—

Configuration Examples for Per-User Configuration

The following sections provide two comprehensive examples:

- TACACS+ Freeware Examples
- RADIUS Examples

These examples show router or access server configuration and AV pair configuration on an AAA server.

TACACS+ Freeware Examples

This section provides the TACACS+ freeware versions of the following examples:

- IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI
- IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI

The following example provides configurations for the TACACS+ freeware daemon, the network access server, and the peer router named Router1. On the TACACS+ AAA server, peer router Router1 has a configuration that includes static routes and IP access lists.

TACACS+ Freeware Daemon Configuration File

```
key = tac123
user = Router1 {
global = cleartext welcome
service = ppp protocol = ip {
route#1=″10.0.0.0 255.0.0.0"
route#2=″10.1.0.0 255.0.0.0"
route#3=″10.2.0.0 255.0.0.0"
inacl#1=″deny 10.5.0.1"
}
}
```

Current Network Access Server Configuration

```
version 11.3
service timestamps debug datetime localtime
service udp-small-servers
service tcp-small-servers
!
hostname Router2
!
aaa new-model
aaa authentication ppp default tacacs+
aaa authorization network tacacs+
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
username Router1 password 7 15050E0007252621
ip host Router2 172.21.114.132
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
interface Ethernet 0
 ip address 172.21.114.132 255.255.255.224
 no ip mroute-cache
 media-type 10BaseT
!

interface Virtual-Template1
 ip unnumbered Ethernet0
 no cdp enable
!
!
interface BRI0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 no ip route-cache
 dialer idle-timeout 300
 dialer map ip 10.5.0.1 name Router1 broadcast 61482
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
!
```

```
ip default-gateway 172.21.114.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.114.129
!
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
tacacs-server host 172.21.114.130
tacacs-server key tac123
```

Current Peer Configuration for Router1

```
version 11.3
no service pad
!
hostname Router1
!
enable secret 5 $1$m1WK$RsjborN1Z.XZuFqsrtSnp/
enable password lab
!
username Router2 password 7 051C03032243430C
ip host Router1 172.21.114.134
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.21.114.134 255.255.255.224
 no ip route-cache
 shutdown
!
interface BRI0
 ip address 10.5.0.1 255.0.0.0
 encapsulation ppp
 dialer map ip 172.21.114.132 name Router2 broadcast 61483
 dialer-group 1
 no fair-queue
!
ip default-gateway 172.21.114.129
no ip classless
ip route 172.21.0.0 255.255.0.0 BRI0
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password lab
 login
end
```

IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

The following example provides configurations for the TACACS+ daemon and the peer router named Router1. On the TACACS+ AAA server, user ny has a configuration that includes inbound and outbound SAP filters.

TACACS+ Freeware Daemon Configuration File for User

```
key = tac123
user = Router1 {
 global = cleartext welcome
 service = ppp protocol = ipx {
        sap=”101 CYBER-01 40.0000.0000.0001 400 10"
        sap=”202 CYBER-02 40.0000.0000.0001 401 10"
        sap=”303 CYBER-03 40.0000.0000.0001 402 10"
        sap-fltr-out#1=”deny 40 101"
        sap-fltr-out#2=”deny 40 202"
        sap-fltr-out#3=”permit -1"
        sap-fltr-in#1=”permit 30 444"
        sap-fltr-in#2=”deny -1"
```

Current Remote Peer (Router1) Configuration

```
version 11.3
!
hostname Router1
!
enable password lab
!
username Router2 password 7 140017070F0B272E
ip host Router1 172.21.114.131
ip name-server 172.19.2.132
ip name-server 192.168.30.32
ipx routing 0000.0c47.090d
ipx internal-network 30
!
interface Ethernet0
 ip address 172.21.114.131 255.255.255.224
!
interface Serial1
 no ip address
 encapsulation ppp
 ipx ipxwan 0 unnumbered peer-Router1
 clockrate 4000000
!
ipx sap 444 ZEON-4 30.0000.0000.0001 444 10
ipx sap 555 ZEON-5 30.0000.0000.0001 555 10
ipx sap 666 ZEON-6 30.0000.0000.0001 666 10
!
Current Network Access Server (Router2) Configuration
version 11.3
service timestamps debug uptime
!
hostname Router2
!
aaa new-model
aaa authentication ppp default tacacs+
aaa authorization network tacacs+
enable password lab
!
username Router1 password 7 044C0E0A0C2E414B
ip host LA 172.21.114.133
ip name-server 192.168.30.32
```

```
ip name-server 172.19.2.132
ipx routing 0000.0c47.12d3
ipx internal-network 40
!
interface Ethernet0
 ip address 172.21.114.133 255.255.255.224
!
interface Virtual-Template1
 no ip address
 ipx ipxwan 0 unnumbered nas-Router2
 no cdp enable
!
interface Serial1
 ip unnumbered Ethernet0
 encapsulation ppp
 ipx ipxwan 0 unnumbered nas-Router2
 ppp authentication chap
!
ipx sap 333 DEEP9 40.0000.0000.0001 999 10
!
virtual-profile virtual-template 1
tacacs-server host 172.21.114.130
tacacs-server key tac123
```

RADIUS Examples

This section provides the RADIUS versions of the following examples:

- IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI
- IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI

The following example shows a remote peer (Router1) configured to dial in to a BRI on a Cisco network access server (Router2), which requests user configuration information from an AAA server (radiusd):

RADIUS User File (Router1)

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:route=10.1.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.2.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.3.0.0 255.0.0.0",
cisco-avpair = "ip:inacl#5=deny 10.0.0.1"
```

Current Network Access Server Configuration

```
version 11.3
service timestamps debug datetime localtime
service udp-small-servers
service tcp-small-servers
!
hostname Router2
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
```

```
!
username Router1 password 7 15050E0007252621
ip host Router2 172.21.114.132
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
interface Ethernet0
 ip address 172.21.114.132 255.255.255.224
 no ip mroute-cache
 media-type 10BaseT
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no cdp enable
!
interface BRI0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 no ip route-cache
 dialer idle-timeout 300
 dialer map ip 10.5.0.1 name Router1 broadcast 61482
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
ip default-gateway 172.21.114.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.114.129
!
virtual-profile vtemplate 1
dialer-list 1 protocol ip permit
radius-server host 172.21.114.130
radius-server key rad123
```

Current Peer Configuration for Router1

```
version 11.3
no service pad
!
hostname Router1
!
enable secret 5 $1$m1WK$RsjborN1Z.XZuFqsrtSnp/
enable password lab
!
username Router2 password 7 051C03032243430C
ip host Router1 172.21.114.134
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.21.114.134 255.255.255.224
 no ip route-cache
 shutdown
!
interface BRI0
 ip address 10.5.0.1 255.0.0.0
 encapsulation ppp
 dialer map ip 172.21.114.132 name Router2 broadcast 61483
 dialer-group 1
 no fair-queue
```

```
!
ip default-gateway 172.21.114.129
no ip classless
ip route 172.21.0.0 255.255.0.0 BRI0
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password lab
 login
!
end
```

Output of ping Command from Router1

```
Router1# ping 172.21.114.132

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.21.114.132, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

(fails due to access list deny)
```

RADIUS Debug Output

```
radrecv: Request from host ac157284 code=1, id=46, length=67
    Client-Id = 172.21.114.132
    Client-Port-Id = 1112670208
    User-Name = "Router1"
    CHAP-Password = "\037\317\213\326*\236)#+\266\243\255x\331\370v\334"
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
Sending Ack of id 46 to ac157284 (172.21.114.132)
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
    [Vendor 9] cisco-avpair = "ip:route=10.0.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:route=10.1.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:route=10.2.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:inacl#5=deny 10.0.0.1"
```

Network Access Server (Router2) show and debug Command Output

```
Router2# show debug

General OS:
  AAA Authorization debugging is on
PPP:
  PPP authentication debugging is on
  Multilink activity debugging is on
ISDN:
  ISDN events debugging is on
Dial on demand:
  Dial on demand events debugging is on
VTEMPLATE:
  Virtual Template debugging is on

pr  4 08:30:09: ISDN BR0: received HOST_INCOMING_CALL
        Bearer Capability i = 0x080010
*Apr  4 08:30:09:       -------------------
        Channel ID i = 0x0101
*Apr  4 08:30:09:         IE out of order or end of 'private' IEs --
        Bearer Capability i = 0x8890
```

```
*Apr  4 08:30:09:         Channel ID i = 0x89
*Apr  4 08:30:09:         Called Party Number i = 0xC1, '61483'
*Apr  4 08:30:09: ISDN BR0: Event: Received a call from <unknown> on B1 at 64 Kb/s
*Apr  4 08:30:09: ISDN BR0: Event: Accepting the call
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
*Apr  4 08:30:09: ISDN BR0: received HOST_CONNECT
        Channel ID i = 0x0101
*Apr  4 08:30:09:       -------------------
        Channel ID i = 0x89
*Apr  4 08:30:09: ISDN BR0: Event: Connected to <unknown> on B1 at 64 Kb/s
*Apr  4 08:30:09: PPP BRI0:1: Send CHAP challenge id=30 to remote
*Apr  4 08:30:10: PPP BRI0:1: CHAP response received from Router1
*Apr  4 08:30:10: PPP BRI0:1: CHAP response id=30 received from Router1
*Apr  4 08:30:10: AAA/AUTHOR/LCP: authorize LCP
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): send AV protocol=lcp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (2084553184): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (2084553184): Post authorization status = PASS_ADD
*Apr  4 08:30:10: PPP BRI0:1: Send CHAP success id=30 to remote
*Apr  4 08:30:10: PPP BRI0:1: remote passed CHAP authentication.
*Apr  4 08:30:10: VTEMPLATE Reuse vaccess1, New Recycle queue size:0

*Apr  4 08:30:10: VTEMPLATE set default vaccess1 with no ip address

*Apr  4 08:30:10: Virtual-Access1 VTEMPLATE hardware address 0000.0c46.154a
*Apr  4 08:30:10: VTEMPLATE vaccess1 has a new cloneblk vtemplate, now it has vtemplate
*Apr  4 08:30:10: VTEMPLATE undo default settings vaccess1

*Apr  4 08:30:10: VTEMPLATE ************* CLONE VACCESS1 ******************Apr  4
08:30:10: VTEMPLATE Clone from vtemplate1 to vaccess1
interface Virtual-Access1
no ip address
encap ppp
ip unnumbered ethernet 0
end

%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Apr  4 08:30:10: AAA/AUTHOR/LCP: authorize LCP
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): send AV protocol=lcp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (1338953760): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (1338953760): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): can we start IPCP?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV protocol=ip
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (1716082074): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (1716082074): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: we can start IPCP (0x8021)
*Apr  4 08:30:10: MLP Bad link Virtual-Access1
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): can we start UNKNOWN?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV protocol=unknown
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (2526612868): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (2526612868): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: we can start UNKNOWN (0x8207)
*Apr  4 08:30:10: MLP Bad link Virtual-Access1
*Apr  4 08:30:10: BRI0:1: Vaccess started from dialer_remote_name
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): can we start IPCP?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV service=ppp
```

```
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV protocol=ip
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (3920403585): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (3920403585): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: we can start IPCP (0x8021)
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): can we start UNKNOWN?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV protocol=unknown
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (3439943223): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (3439943223): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: we can start UNKNOWN (0x8207)
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: start: her address 10.0.0.1, we want
0.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV servi*Apr  4 08:30:13:
AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV protocol=ip
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV addr*10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (3215797579): Method=RADIUS
*Apr  4 08:30:13: AAA/AUTHOR (3215797579): Post authorization status = PASS_ADD
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV protocol=ip
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV addr*10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.1.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.2.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.3.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV inacl#5=deny 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: authorization succeeded
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: done: her address 10.0.0.1, we want
10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: authorization succeeded
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 10.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 10.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 11.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 11.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 12.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 12.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: parse 'ip access-list standard Virtual-Access1#1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: parse 'deny 10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip access-list
standard Virtual-Access1#1
*Apr  4 08:30:13: VTEMPLATE vaccess1 has a new cloneblk AAA, now it has vtemplate/AAA
*Apr  4 08:30:13: VTEMPLATE ************* CLONE VACCESS1 *****************

*Apr  4 08:30:13: VTEMPLATE Clone from AAA to vaccess1
interface Virtual-Access1
ip access-group Virtual-Access1#1 in

*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: vaccess parse 'interface Virtual-Access1
ip access-group Virtual-Access1#1 in
' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Check for unauthorized mandatory AV's
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Processing AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Processing AV protocol=unknown
*Apr  4 08:30:13: AAA/AUTHOR/FSM: succeeded
%ISDN-6-CONNECT: Interface BRI0:1 is now connected to Router1
```

```
Router2# show ip access-list

Standard IP access list Virtual-Access1#1 (per-user)
    deny   10.0.0.1

Router2# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.21.114.129 to network 0.0.0.0

U    10.0.0.0/8 [1/0] via 10.3.0.1
U    10.1.0.0/8 [1/0] via 10.3.0.1
U    10.2.0.0/8 [1/0] via 10.3.0.1
     10.3.0.0/8 is subnetted, 1 subnets
C       10.3.0.1 is directly connected, Virtual-Access1
     172.21.0.0/16 is subnetted, 1 subnets
C       172.21.114.128 is directly connected, Ethernet0
S*   0.0.0.0/0 [1/0] via 172.21.114.129


Router2# show interfaces virtual-access 1

Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered.  Using address of Ethernet0 (172.21.114.132)
  MTU 1500 bytes, BW 64 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open, multilink Closed
  Open: IPCP, CDP
  Last input 5d04h, output never, output hang never
  Last clearing of "show interface" counters 00:06:42
  Queueing strategy: fifo
  Output queue 0/40, 0 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
     76 packets input, 3658 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     141 packets output, 2909 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions

Router2# show ip interface virtual-access 1

Virtual-Access1 is up, line protocol is up
  Interface is unnumbered.  Using address of Ethernet0 (172.21.114.132)
  Broadcast address is 255.255.255.255
  Peer address is 10.0.0.1
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Outgoing access list is not set
  Inbound  access list is Virtual-Access1#1
  Proxy ARP is enabled
  Security level is default
```

```
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled


Router2# debug ip packet

IP packet debugging is on
Router2#
*Apr  4 08:30:42: IP: s=172.21.114.129 (Ethernet0), d=255.255.255.255, len 186, rcvd 2
*Apr  4 08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, a*Apr  4
08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access denied
*Apr  4 08:30:42: IP: s=172.21.114.132 (local), d=10.0.0.1 (Virtual-Access1), len 4,
sending
*Apr  4 08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
*Apr  4 08:30:44: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
*Apr  4 08:30:44: IP: s=172.21.114.132 (local), d=10.0.0.1 (Virtual-Access1), len 16,
sending
*Apr  4 08:30:44: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
```

IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

The following examples show a remote peer (Router1) configured to dial in to a synchronous interface on a Cisco network access server (Router2), which requests user configuration information from an AAA server (radiusd):

RADIUS User File (Router 1)

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ipx:sap=101 CYBER-01 40.0000.0000.0001 400 10",
cisco-avpair = "ipx:sap=202 CYBER-02 40.0000.0000.0001 401 10",
cisco-avpair = "ipx:sap=303 CYBER-03 40.0000.0000.0001 402 10",
cisco-avpair = "ipx:sap-fltr-out#20=deny 40 101",
cisco-avpair = "ipx:sap-fltr-out#21=deny 40 202",
cisco-avpair = "ipx:sap-fltr-out#22=permit -1",
cisco-avpair = "ipx:sap-fltr-in#23=permit 30 444",
cisco-avpair = "ipx:sap-fltr-in#23=deny -1"
```

Current Remote Peer (Router 1) Configuration

```
hostname Router1
!
enable password lab
!
username Router2 password 7 140017070F0B272E
ip host Router1 172.21.114.131
ip name-server 172.19.2.132
ip name-server 192.168.30.32
ipx routing 0000.0c47.090d
ipx internal-network 30
!
interface Ethernet0
 ip address 172.21.114.131 255.255.255.224
!
```

```
interface Serial1
 no ip address
 encapsulation ppp
 ipx ipxwan 0 unnumbered peer-Router1
 clockrate 4000000
!
ipx sap 444 ZEON-4 30.0000.0000.0001 444 10
ipx sap 555 ZEON-5 30.0000.0000.0001 555 10
ipx sap 666 ZEON-6 30.0000.0000.0001 666 10
!
...
version 12.1
service timestamps debug uptime
!
hostname Router2
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable password lab
!
username Router1 password 7 044C0E0A0C2E414B
ip host Router2 172.21.114.133
ip name-server 172.22.30.32
ip name-server 192.168.2.132
ipx routing 0000.0c47.12d3
ipx internal-network 40
!
interface Ethernet0
 ip address 172.21.114.133 255.255.255.224
!
interface Virtual-Template1
 no ip address
 ipx ipxwan 0 unnumbered nas-Router2
 no cdp enable
!
interface Serial1
 ip unnumbered Ethernet0
 encapsulation ppp
 ipx ipxwan 0 unnumbered nas-Router2
 ppp authentication chap
!
ipx sap 333 DEEP9 40.0000.0000.0001 999 10
!
virtual-profile vtemplate 1
radius-server host 172.21.114.130
radius-server key rad123
```

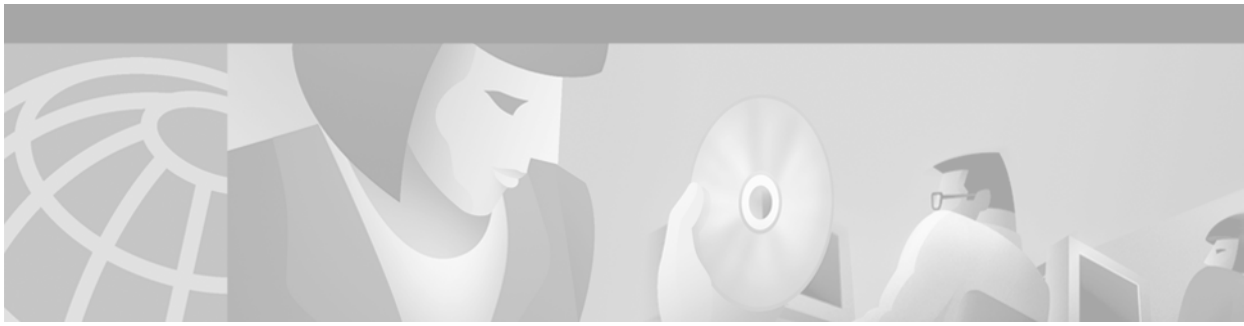
RADIUS debug Output

```
radrecv: Request from host ac157285 code=1, id=23, length=67
    Client-Id = 172.21.114.133
    Client-Port-Id = 1399128065
    User-Name = "Router1"
    CHAP-Password = "%"(\012I$\262\352\031\276\024\302\277\225\347z\274"
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
Sending Ack of id 23 to ac157285 (172.21.114.133)
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
    [Vendor 9] cisco-avpair = "ipx:sap=101 CYBER-01 40.0000.0000.0001 400 10"
    [Vendor 9] cisco-avpair = "ipx:sap=202 CYBER-02 40.0000.0000.0001 401 10"
    [Vendor 9] cisco-avpair = "ipx:sap=303 CYBER-03 40.0000.0000.0001 402 10"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#20=deny1 40 101"
```

```
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#21=deny 40 202"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#22=permit -1"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-in#23=permit 30 444"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-in#23=deny -1"
```

Network Access Server show Command Output

```
Router2# show ipx servers

Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail
5 Total IPX Servers

Table ordering is based on routing and server info

   Type Name                          Net     Address    Port     Route Hops Itf
s   101 CYBER-01                      40.0000.0000.0001:0400      conn  10  Int
s   202 CYBER-02                      40.0000.0000.0001:0401      conn  10  Int
s   303 CYBER-03                      40.0000.0000.0001:0402      conn  10  Int
S   333 DEEP9                         40.0000.0000.0001:0999      conn  10  Int
P   444 ZEON-4                        30.0000.0000.0001:0444      7/01  11  Vi1


Router1# show ipx servers

Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail
5 Total IPX Servers

Table ordering is based on routing and server info

   Type Name                          Net     Address    Port     Route Hops Itf
P   303 CYBER-03                      40.0000.0000.0001:0402      7/01  11  Se1
P   333 DEEP9                         40.0000.0000.0001:0999      7/01  11  Se1
S   444 ZEON-4                        30.0000.0000.0001:0444      conn  10  Int
S   555 ZEON-5                        30.0000.0000.0001:0555      conn  10  Int
S   666 ZEON-6                        30.0000.0000.0001:0666      conn  10  Int

Router2# show ipx access-list

IPX sap access list Virtual-Access1#2
    permit 30 444
    deny FFFFFFFF
IPX sap access list Virtual-Access1#3
    deny 40 101
    deny 40 202
    permit FFFFFFFF
```



Configuring Resource Pool Management

This chapter describes the Cisco Resource Pool Management (RPM) feature. It includes the following main sections:

To identify the hardware platform or software image information associated with a feature, 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.

For a complete description of the commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*, Release 12.2. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

RPM Overview

Cisco RPM enables telephone companies and Internet service providers (ISPs) to share dial resources for wholesale and retail dial network services. With RPM, telcos and ISPs can count, control, and manage dial resources and provide accounting for shared resources when implementing different service-level agreements.

You can configure RPM in a single, standalone Cisco network access server (NAS) by using RPM or, optionally, across multiple NAS stacks by using one or more external Cisco Resource Pool Manager Servers (RPMS).

Cisco RPM gives data network service providers the capability to do the following:

- Have the flexibility to include local retail dial services in the same NAS with the wholesale dial customers.
- Manage customer use of shared resources such as modems or High-Level Data Link Control (HDLC) controllers for data calls.
- Offer advanced wholesale dialup services using a Virtual Private Dialup Network (VPDN) to enterprise accounts and ISPs.
- Deploy Data over Voice Bearer Service (DoVBS).

- Manage call sessions by differentiating dial customers through customer profiles. The customer profile determines where resources are allocated and is based on the incoming Dialed Number Information Service (DNIS) number or Calling Line Identification (CLID).
- Efficiently use resource groups such as modems to offer differing over subscription rates and dial service-level agreements.

Note Ear and Mouth Feature Group B (E&M-FGB) is the only signaling type supported for channel-associated signaling (CAS) on T1 and T3 facilities; R2 is supported for E1 facilities. FG D is not supported. Cisco IOS software collects DNIS digits for the signaling types FGB, PRI, and SS7 and only E&M-FGB and R2 CAS customer profiles are supported. For all other CAS signaling types, use the default DNIS group customer profiles.

Components of Incoming and Outgoing Call Management

Cisco RPM manages both incoming calls and outgoing sessions. Cisco RPM differentiates dial customers through configured customer profiles based on the DNIS and call type determined at the time of an incoming call.

The components of incoming call management in the Cisco RPM are described in the following sections:

- Customer Profile Types
- DNIS Groups
- Call Types
- Resource Groups
- Resource Services

You can use Cisco RPM to answer all calls and differentiate customers by using VPDN profiles and groups. The components of outgoing session management in the Cisco RPM are described in the following sections:

- VPDN Groups
- VPDN Profiles

Note These components of Cisco RPM are enabled after the NAS and other equipment has been initially set up, configured, and verified for proper operation of the dial, PPP, VPDN, and authentication, authorization, and accounting (AAA) segments. Refer to the Cisco IOS documentation for these other segments for installation, configuration, and troubleshooting information before attempting to use RPM.

Configured DNIS groups and resource data can be associated to customer profiles. These customer profiles are selected by the incoming call DNIS number and call type and then used to identify resource allocations based on the associated resource groups and defined resource services.

After the call is answered, customer profiles can also be associated with VPDN groups so the configured VPDN sessions and other data necessary to set up or reject a VPDN session are applied to the answered calls. VPDN group data includes associated domain name or DNIS, IP addresses of endpoints, maximum sessions per endpoint, maximum Multilink PPP (MLP) bundles per VPDN group, maximum links per MLP bundle, and other tunnel information.

Customer Profile Types

There are three types of customer profiles in Cisco RPM, which are described in the following sections:

- Customer Profiles
- Default Customer Profiles
- Backup Customer Profiles

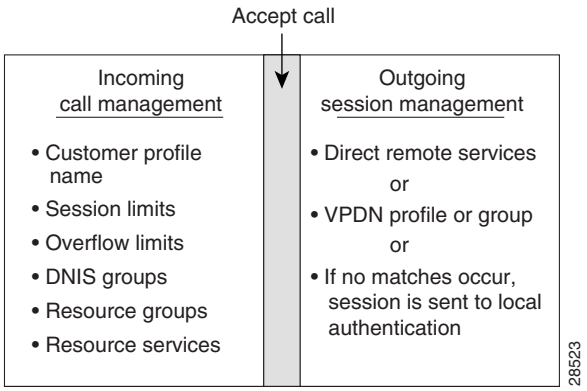
Additionally, you can create a customer profile template and associate it with a customer profile; it is then integrated into the customer profile.

Customer Profiles

A customer profile defines how and when to answer a call. Customer profiles include the following components (see Figure 99):

- Customer profile name and description—Name and description of the customer.
- Session limits—Maximum number of standard sessions.
- Overflow limits—Maximum number of overflow sessions.
- DNIS groups.
- CLID.
- Resource groups.
- Resource services.
- VPDN groups and VPDN profiles.
- Call treatment—Determines how calls that exceed the session and overflow limits are treated.

Figure 99 Components of a Customer Profile



The incoming side of the customer profile determines if the call will be answered using parameters such as DNIS and call type from the assigned DNIS group and session limits. The call is then assigned the appropriate resource within the resource group defined in the customer profile. Each configured customer profile includes a maximum allowed session value and an overflow value. As sessions are started and ended, session counters are incremented and decremented so customer status is kept current. This information is used to monitor the customer resource limit and determine the appropriate call treatment based on the configured session limits.

The outgoing side of the customer profile directs the answered call to the appropriate destination:

- To a local AAA server of retail dial applications and Internet/intranet access.
- To a tunnel that is established between the NAS or L2TP Access Concentrator (LAC) to a wholesale VPDN home gateway of a dial customer, or L2TP Network Server (LNS) using Layer 2 Forwarding Protocol (L2F) or Layer 2 Tunneling Protocol (L2TP) technology.

Default Customer Profiles

Default customer profiles are identical to standard customer profiles, except that they do not have any associated DNIS groups. Default customer profiles are created using the reserved keyword **default** for the DNIS group.

Default customer profiles are used to provide session counting and resource assignment to incoming calls that do not match any of the configured DNIS groups. Although specific resources and DNIS groups can be assigned to customer profiles, default customer profiles allow resource pooling for the calls that do not match the configured DNIS groups or where the DNIS is not provided. Retail dial services and domain-based VPDN use default customer profiles.

When multiple default customer profiles are used, the call type (speech, digital, V.110, or V.120) of the default DNIS group is used to identify which default customer profile to use for an incoming call. At most, four default profiles (one for each call type) can be configured.

Note If default customer profiles are not defined, then calls that do not match a DNIS group in a customer profile are rejected with a "no answer" or "busy" call treatment sent to the switch.

Backup Customer Profiles

Backup customer profiles are customer profiles configured locally on the Cisco NAS and are used to answer calls based on a configured allocation scheme when the link between the Cisco NAS and Cisco RPMS is disabled. See the section "Configuring Customer Profiles Using Backup Customer Profiles" for more information about configuring backup customer profiles.

Customer Profile Template

With RPM, users can also implement wholesale dial services without using VPDN tunnels to complete dial-in calls to destinations of the end customer. This capability is accomplished with components of the AAA groups and the PPP configurations.

The AAA group provides IP addresses of AAA servers for authentication and accounting. The PPP configurations allow users to configure the Cisco IOS PPP feature set on each customer profile. In this current implementation, PPP configuration is based on the following:

- Applicable IP address pool(s) or default local list of IP addresses
- Primary and secondary Domain Name System (DNS) or Windows Internet naming service (WINS)
- Number of links allowed for each call using MLP

Note The AAA and PPP integration applies to a single NAS environment.

To add PPP configurations to a customer profile, you must create a customer profile template. Once you create the template and associate it with a customer profile using the **source template** command, it is integrated into the customer profile.

The RPM customer profile template for the PPP command set, when used with the Cisco IOS feature, Server Groups Selected by DNIS, presents a strong single NAS solution for providers of wholesale dial services, as follows:

- Call acceptance is determined by the RPM before call answering, using the configured size limits and resource availability.
- The answered call then uses the PPP configuration defined in the template to initiate authentication, obtain an IP address, and select a DNS or WINS that is located at the customer site.
- The same DNIS that was used to choose the customer profile selects the servers for authentication/authorization and accounting that are located at the wholesale customer's site.

The section "Configuring a Customer Profile Template" later in this chapter describes how to create a customer profile template so that you can configure the Cisco IOS PPP features on a customer profile, but this section does not list the existing PPP command set. For information about the PPP command set, refer to the *Cisco IOS Dial Technologies Command Reference*.

DNIS Groups

A DNIS group is a configured list of DNIS called party numbers that correspond to the numbers dialed to access particular customers, service offerings, or both. For example, if a customer from phone number 000-1234 calls a number 000-5678, the DNIS provides information on the number dialed—000-5678.

Cisco RPM checks the DNIS number of inbound calls against the configured DNIS groups, as follows:

- If Cisco RPM finds a match, it uses the configured information in the customer profile to which the DNIS group is assigned.
- If Cisco RPM does not find a match, it uses the configured information in the customer profile to which the default DNIS group is assigned.
- The DNIS/call type sequence can be associated only with one customer profile.

CLID Groups

A CLID group is a configured list of CLID calling party numbers. The CLID group specifies a list of numbers to reject if the group is associated with a call discriminator. For example, if a customer from phone number 000-1234 calls a number 000-5678, the CLID provides information on the calling party number—000-1234.

A CLID can be associated with only one CLID group.

Call Types

Call types from calls originating from ISDN, SS7, and CAS (CT1, CT3, and CE1) are used to assign calls to the appropriate resource. Call types for ISDN and SS7 are based on Q.931 bearer capability. Call types for CAS are assigned based on static channel configuration.

Supported call types are as follows:

- Speech
- Digital
- V.110
- V.120

Note Voice over IP, fax over IP, and dial-out calls are not supported in RPM.

Resource Groups

Cisco RPM enables you to maximize the use of available shared resources within a Cisco NAS for various resource allocation schemes to support service-level agreements. Cisco RPM allows you to combine your Cisco NAS resource groups with call types (speech, digital, V.110, and V.120) and optional resource modem services. Resource groups and services are configured for customer profiles and assigned to incoming calls through DNIS groups and call types.

Resource groups have the following characteristics:

- Are configured on the Cisco NAS and applied to a customer profile.
- Represent groupings of similar hardware or firmware that are static and do not change on a per-call basis.
- Can define resources that are port-based or not port-based:
  - Port-based resources are identified by physical location, such as a range of port/slot numbers (for example, modems or terminal adapters).
  - Non-port-based resources are identified by a single size parameter (for example, HDLC framers or V.120 terminal adapters—V.120 terminal adapters are currently implemented as part of Cisco IOS software).

Resource assignments contain combinations of Cisco NAS resource groups, optional resource modem services, and call types. The NAS resources in resource groups that have not been assigned to a customer profile will not be used.

Note To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource. The resource group assigned to this customer profile will be "digital resources" and also have a call type of "speech," so the call will terminate on an HDLC controller rather than a modem.

Resource Services

A resource service contains a finite series of resource command strings that can be used to help dynamically configure an incoming connection. Services supported by a resource group are determined by the combination of hardware and firmware installed. Currently, resource service options can be configured and applied to resource groups. Resource services can be defined to affect minimum and maximum speed, modulation, error correction, and compression, as shown in Table 41.

Table 41 Resource Services

Service	Options	Comments
min-speed	<300–56000>, any	Must be a V.90 increment.
max-speed	<300–56000>, any	Must be a V.90 increment.
modulation	k56flex, v22bis, v32bis, v34, v90, any	None.

Table 41 Resource Services (continued)

Service	Options	Comments
error-correction	1apm, mn14	This is a hidden command.
compression	mnps, v42bis	This is a hidden command.

VPDN Groups

The VPDN group contains the data required to build a VPDN tunnel from the RPM NAS LAC to the LNS. In the context of RPM, VPDN is authorized by first associating a customer profile with a VPDN group, and second by associating the VPDN group to the DNIS group used for that customer profile. VPDN group data includes the endpoint IP addresses.

Cisco RPM enables you to specify multiple IP endpoints for a VPDN group, as follows:

- If two or more IP endpoints are specified, Cisco RPM uses a load-balancing method to ensure that traffic is distributed across the IP endpoints.
- For DNIS-based VPDN dial service, VPDN groups are assigned to customer profiles based on the incoming DNIS number and the configured DNIS groups.
- For domain-based VPDN dial service, VPDN groups are assigned to the customer profile or the default customer profile with the matching call-type assignment.
- For either DNIS-based or domain-based VPDN dial services, there is a customer profile or default customer profile for the initial resource allocation and customer session limits.

The VPDN group provides call management by allowing limits to be applied to both the number of MLP bundles per tunnel and the number of links per MLP bundle. Limits can also restrict the number of sessions per IP endpoint. If you require more granular control of VPDN counters, use VPDN profiles.

VPDN Profiles

VPDN profiles allow session and overflow limits to be imposed for a particular customer profile. These limits are unrelated to the limits imposed by the customer profile. A customer profile is associated with a VPDN profile. A VPDN profile is associated with a VPDN group. VPDN profiles are required only when these additional counters are required for VPDN usage per customer profile.

Call Treatments

Call treatment determines how calls are handled when certain events require the call to be rejected. For example, if the session and overflow limits for one of your customers have been exceeded, any additional calls will receive a busy signal (see Table 42).

Table 42 Call-Treatment Table

Event	Call-Treatment Option	Results
Customer profile not found	No answer (default)	The caller receives rings until the switch eventually times out. Implies that the NAS was appropriate, but resources were unavailable. The caller should try later.
	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller. The call is rejected based on not matching a DNIS group/call type and customer profile. Can be used to immediately reject the call and free up the circuit.
Customer profile limits exceeded	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller.
NAS resource not available	Channel not available (default)	The switch sends the call to the next channel in the trunk group. The call can be answered, but the NAS does not have any available resources in the resource groups. Allows the switch to try additional channels until it gets to a different NAS in the same trunk group that has the available resources.
	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller. Can be used when the trunk group does not span additional NASes.
Call discrimination match	No answer	The caller receives rings until the switch eventually times out.

Details on RPM Call Processes

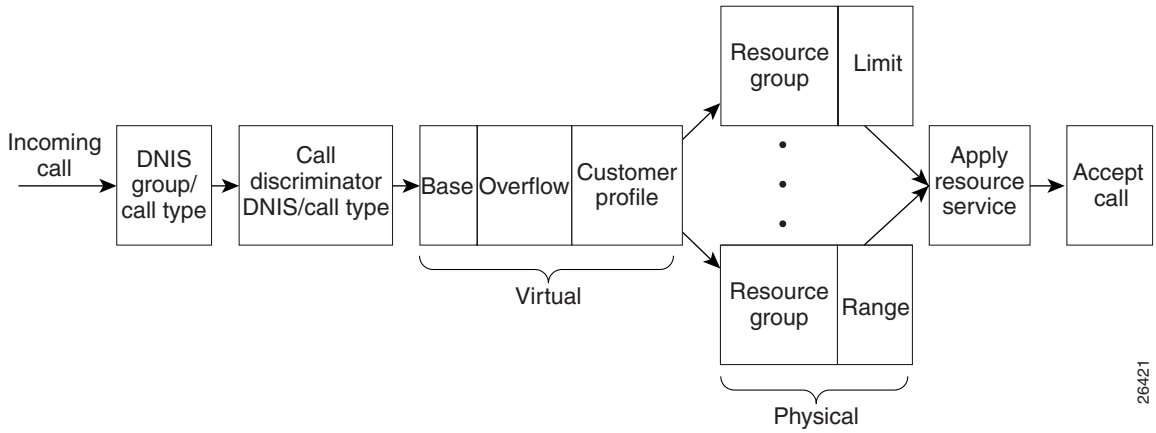
On the incoming call management of the customer profile, the following sequence occurs to determine if a call is answered:

1. The incoming DNIS is mapped to a DNIS group; if there is no incoming DNIS number, or the DNIS number provided does not match any configured DNIS group, the DNIS group *default* is used.
2. The mapped DNIS group is checked against configured call discriminator profiles to confirm if this DNIS group/call-type combination is disallowed. If there is a match, the call is immediately rejected.
3. Once a DNIS group or a default DNIS group is identified, the customer profile associated with that DNIS group and the call type (from the bearer capability for ISDN call, statically configured for CAS calls) is selected. If there is no corresponding customer profile, the call is rejected.
4. The customer profile includes a session limit value and an overflow limit value. If these thresholds are not met, the call is then assigned the appropriate resource defined in the customer profile. If the thresholds are met, the call is rejected.

5. If resources are available from the resource group defined in the customer profile, the call is answered. Otherwise, the call is rejected.
6. As sessions start and end, the session counters increase and decrease, so the customer profile call counters are kept current.

See Figure 100 for a graphical illustration of the RPM call processes.

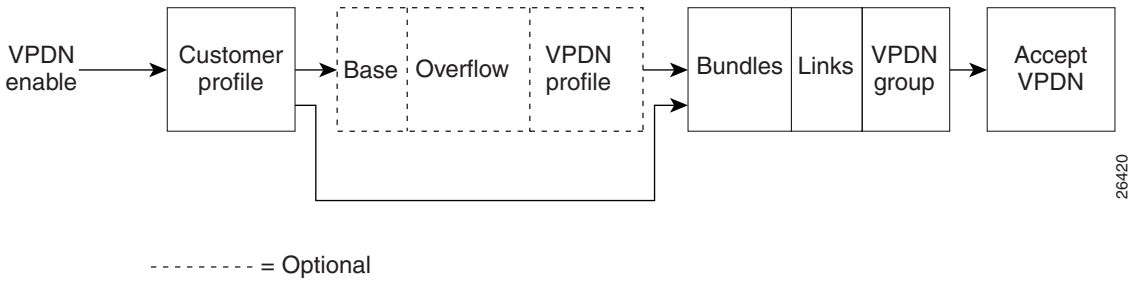
Figure 100 Incoming Call Management: RPM Functional Description



After the call is answered and if VPDN is enabled, Cisco RPM checks the customer profile for an assigned VPDN group or profile. The outgoing session management of the customer profile directs the answered call to the appropriate destination (see Figure 101), as follows:

- To a local AAA server of retail dial applications and Internet/intranet access.
- To a tunnel that is established between the NAS or LAC and a wholesale VPDN home gateway from a dial customer or LNS using L2F or L2TP tunneling technology.

Figure 101 Outgoing Call Management: RPM Functional Description for VPDN Profiles and Groups



If a VPDN profile is found, the limits are checked, as follows:

- If the limits have not been exceeded, the VPDN group data associated with that VPDN profile is used to build a VPDN tunnel.
- If the VPDN limits have been exceeded, the call is disconnected.

If a VPDN group is found within the customer profile, the VPDN group data is used to build a VPDN tunnel, as follows:

- If the VPDN group limits (number of multilink bundles, number of links per bundle) have not been exceeded, a VPDN tunnel is built.
- If the limits have been reached, the call is disconnected.

If no VPDN profile is assigned to the customer profile and VPDN is enabled, non-RPM VPDN service is attempted. If the attempt fails, the call is processed as a retail dial service call if local AAA service is available.

Accounting Data

You can generate accounting data for network dial service usage in NAS AAA attribute format.

You can configure the Cisco NAS to generate AAA accounting records for access to external AAA server option. The accounting start and stop records in AAA attribute format are sent to the external AAA server using either RADIUS server hosts or TACACS+ protocols for accounting data storage. Table 43 lists the new fields in the AAA accounting packets.

Table 43 AAA Accounting Records

Accounting Start Record	Accounting Stop Record
Call-Type	Disconnect-Cause
CAS-Group-Name	Modem-Speed-Receive
Customer-Profile-Name	Modem-Speed-Transmit
Customer-Profile-Active-Sessions	MLP-Session-ID
DNIS-Group-Name	
Overflow	
MLP-Session_ID	
Modem-Speed-Receive	
Modem-Speed-Transmit	
VPDN-Domain-Name	
VPDN-Tunnel-ID	
VPDN-HomeGateway	
VPDN-Group-Active-Sessions	

Data over Voice Bearer Services

DoVBS is a dial service that uses a customer profile and an associated resource group of digital resources to direct data calls with a speech call type to HDLC controllers.

To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource.

The resource group assigned to this customer profile will be “digital resources” and will also have a call type of speech, so the call will terminate on an HDLC controller rather than a modem.

Call Discriminator Profiles

The Cisco RPM CLID/DNIS Call Discriminator feature lets you specify a list of calling party numbers to be rejected for inbound calls. This Cisco IOS Release 12.2 CLID/DNIS call screening feature expands previous call screening features in Cisco RPM. CLID/DNIS call screening provides an additional way to screen calls on the basis of CLID/DNIS for both local and remote RPM.

Cisco RPM CLID/DNIS Call Discriminator profiles enable you to process calls differently on the basis of the call type and CLID combination. Resource pool management offers a call discrimination feature that rejects calls on the basis of a CLID group and a call type filter. When a call arrives at the NAS, the CLID and the call type are matched against a table of disallowed calls. If the CLID and call type match entries in this table, the call is rejected before it is assigned Cisco NAS resources or before any other Cisco RPM processing occurs. This is called precall screening.

Precall screening decides whether the call is allowed to be processed. You can use the following types of discriminators to execute precall screening:

- ISDN discriminator—Accepts a call if the calling number matches a number in a group of configured numbers (ISDN group). This is also called white box screening. If you configure an ISDN group, only the calling numbers specified in the group are accepted.
- DNIS discriminator—Accepts a call if the called party number matches a number in a group of configured numbers (DNIS group). If you set up a DNIS group, only the called party numbers in the group are accepted. DNIS gives you information about the called party.
- Cisco RPM CLID/DNIS discriminator—Rejects a call if the calling number matches a number in a group of configured numbers (CLID/DNIS group). This is also called black box screening.

  If you configure a discriminator with a CLID group, the calling party numbers specified in the group are rejected. CLID gives you information about the caller.

  Similarly, if you configure a discriminator with a DNIS group, the called party numbers specified in the group are rejected.

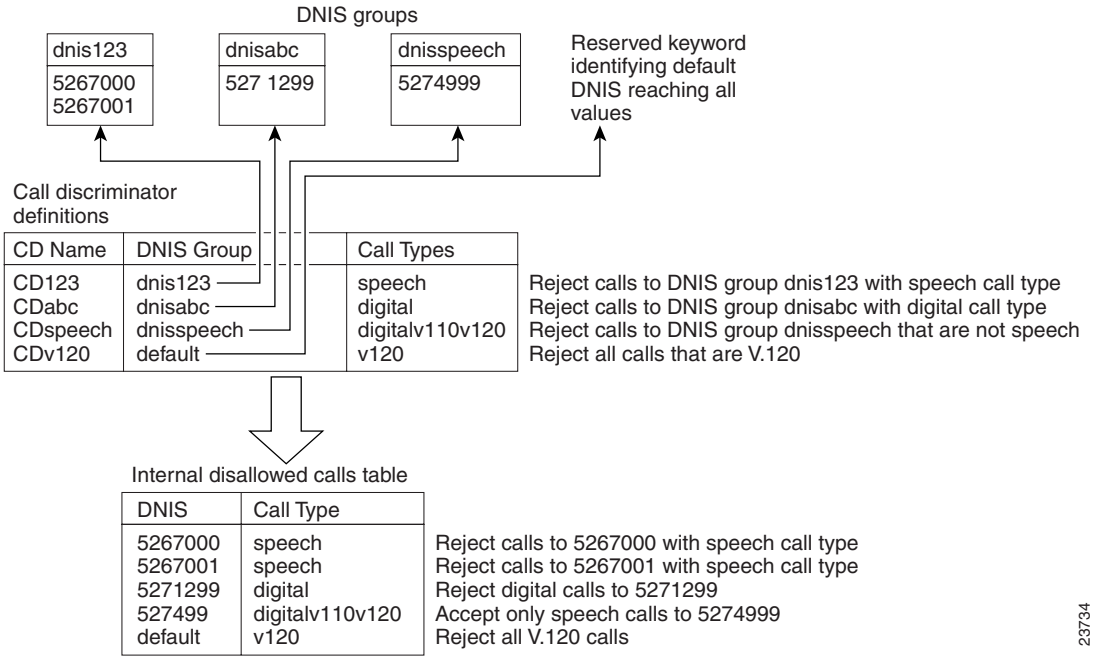
The Cisco RPM CLID/DNIS Call Discriminator Feature is independent of ISDN or DNIS screening done by other subsystems. ISDN or DNIS screening and Cisco RPM CLID/DNIS screening can both be present in the same system. Both features are executed if configured. Similarly, if DNIS Preauthorization using AAA is configured, it is present in addition to Cisco RPM CLID/DNIS screening. Refer to the *Cisco IOS Security Configuration Guide* for more information about call preauthorization.

In Cisco RPM CLID/DNIS screening, the discriminator can be a CLID discriminator, a DNIS discriminator, or a discriminator that screens on both the CLID and DNIS. The resulting discrimination logic is:

- If a discriminator contains just DNIS groups, it is a DNIS discriminator that ignores CLID. The DNIS discriminator blocks the call if the called number is in a DNIS group, which the call type references.
- If a discriminator contains just CLID groups, it is a CLID discriminator that ignores DNIS. The CLID discriminator blocks the call if the calling number is in a CLID group, which the call type references.
- If a discriminator contains both CLID and DNIS groups, it is a logical AND discriminator. It blocks the call if the calling number and called number are in the CLID or DNIS group, and the call type references the corresponding discriminator.

Figure 102 shows how call discrimination can be used to restrict a specific DNIS group to only modem calls by creating call discrimination settings for the DNIS group and the other supported call types (digital, V.110, and V.120).

Figure 102 Call Discrimination



Incoming Call Preauthentication

With ISDN PRI or channel-associated signaling (CAS), information about an incoming call is available to the NAS before the call is connected. The available call information includes:

- The DNIS, also referred to as the *called number*
- The CLID, also referred to as the *calling number*
- The call type, also referred to as the *bearer capability*

The Preauthentication with ISDN PRI and Channel-Associated Signalling feature introduced in Cisco IOS Release 12.2 allows a Cisco NAS to decide—on the basis of the DNIS number, the CLID number, or the call type—whether to connect an incoming call.

When an incoming call arrives from the public network switch, but before it is connected, this feature enables the NAS to send the DNIS number, CLID number, and call type to a RADIUS server for authorization. If the server authorizes the call, the NAS accepts the call. If the server does not authorize the call, the NAS sends a disconnect message to the public network switch to reject the call.

The Preauthentication with ISDN PRI and Channel-Associated Signalling feature offers the following benefits:

- With ISDN PRI, it enables user authentication and authorization before a call is answered. With CAS, the call must be answered; however, the call can be dropped if preauthentication fails.
- It enables service providers to better manage ports using their existing RADIUS solutions.
- Coupled with a preauthentication RADIUS server application, it enables service providers to efficiently manage the use of shared resources to offer differing service-level agreements.

For more information about the Preauthentication with ISDN PRI and Channel-Associated Signalling feature, refer to the *Cisco IOS Security Configuration Guide*.

RPM Standalone Network Access Server

A single NAS using Cisco RPM can provide the following:

- Wholesale VPDN dial service to corporate customers
- Direct remote services
- Retail dial service to end users

Figure 103 and Figure 104 show multiple connections to a Cisco AS5300 NAS. Incoming calls to the NAS can use ISDN PRI signaling, CAS, or the SS7 signaling protocol. Figure 103 shows incoming calls that are authenticated locally for retail dial services or forwarded through VPDN tunnels for wholesale dial services.

Note This implementation does not use Cisco RPM CLID/DNIS Call Discriminator Feature. If you are not using Cisco RPMS and you have more than one Cisco NAS, you must manually configure each NAS by using Cisco IOS commands. Resource usage information is not shared between NASes.

Figure 103 Retail Dial Service Using RPM

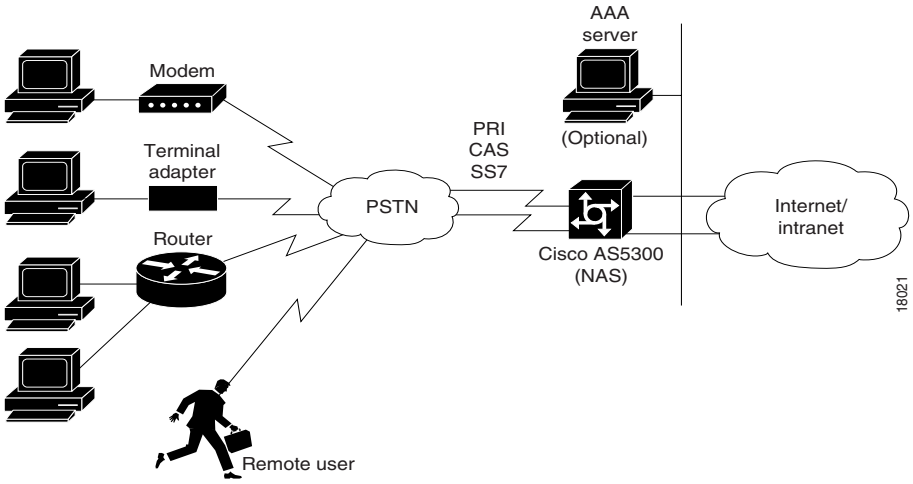
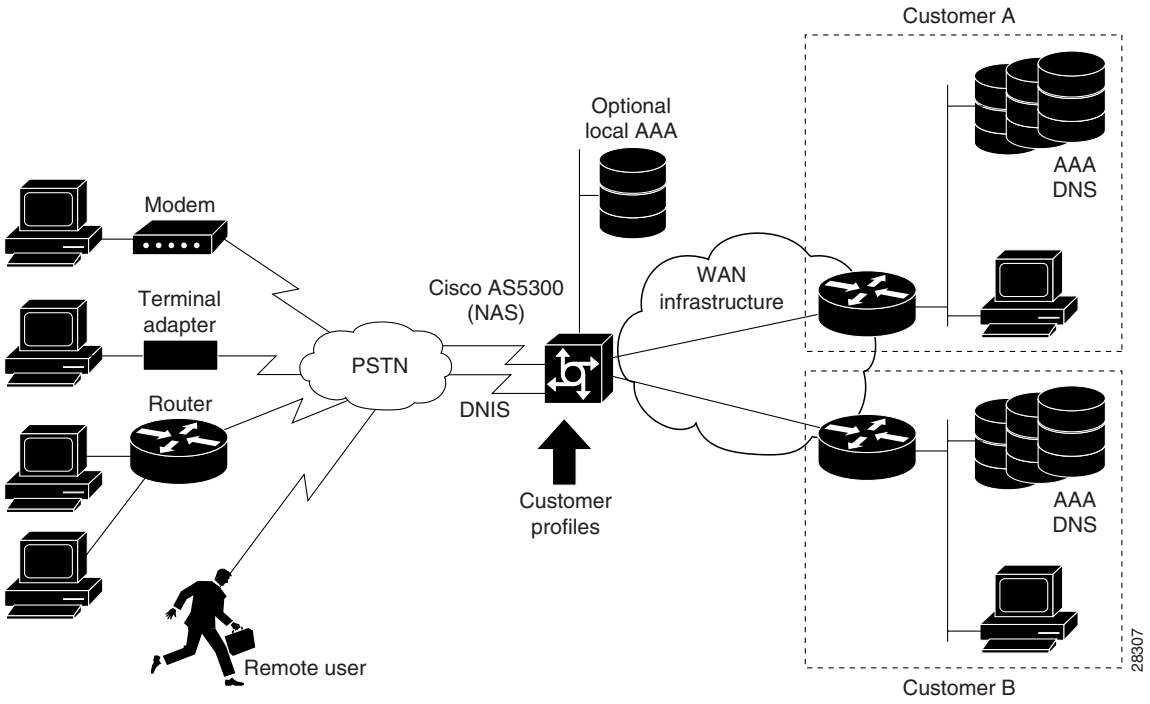


Figure 104 shows a method of implementing wholesale dial services without using VPDN tunnels by creating individual customer profiles that consist of AAA groups and PPP configurations. The AAA groups provide IP addresses of AAA servers for authentication and accounting. The PPP configurations enable you to set different PPP parameter values on each customer profile. A customer profile typically includes the following PPP parameters:

- Applicable IP address pools or a default local list of IP addresses
- Primary and secondary DNS or WINS
- Authentication method such as the Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), or Microsoft CHAP Version 1 (MS-CHAP)
- Number of links allowed for each call using Multilink PPP

Note The AAA and PPP integration applies to a single NAS environment; the external RPMS solution is not supported.

Figure 104 Resource Pool Management with Direct Remote Services



Call Processing

For call processing, incoming calls are matched to a DNIS group and the customer profile associated with that DNIS group. If a match is found, the customer profile session and overflow limits are applied and if available, the required resources are allocated. If a DNIS group is not found, the customer profile associated with the default DNIS group is used. The call is rejected if a customer profile using the default DNIS group cannot be found.

After the call is answered and if VPDN is enabled, the Cisco RPM checks the customer profile for an assigned VPDN group or profile. If a VPDN group is found, Cisco RPM authorizes VPDN by matching the group domain name or DNIS with the incoming call. If a match is found, VPDN profile session and overflow limits are applied, and, if the limits are not exceeded, tunnel negotiation begins. If the VPDN limits are exceeded, the call is disconnected.

If no VPDN profile is assigned to the customer profile and VPDN is enabled, non-RPM VPDN service will be attempted. If it fails, the call is processed as a retail dial service call if local AAA service is available.

Base Session and Overflow Session Limits

Cisco RPM enables you to set base and overflow session limits in each customer profile. The base session limit determines the maximum number of nonoverflow sessions supported for a customer profile. When the session limit is reached, if overflow sessions are not enabled, any new calls are rejected. If overflow sessions are enabled, new sessions up to the session overflow limit are processed and marked as overflow for call handling and accounting.

The session overflow limit determines the allowable number of sessions above the session limit. If the session overflow limit is greater than zero, overflow sessions are enabled and the maximum number of allowed sessions is the session limit plus the session overflow limit. While the session overflow limit has been reached, any new calls are rejected. Table 44 summarizes the effects of session and session overflow limits.

Enabling overflow sessions is useful for allocating extra sessions for preferred customers at premium rates. Overflow sessions can also be useful for encouraging customers to adequately forecast bandwidth usage or for special events when normal session usage is exceeded. For example, if a customer is having a corporate-wide program and many people are expected to request remote access, you could enable many overflow sessions and charge a premium rate for the excess bandwidth requirements.

Note An overflow call is a call received while the session limit is exceeded and is in an overflow state. When a call is identified as an overflow call, the call maintains the overflow status throughout its duration, even if the number of current sessions returns below the session limit.

Table 44 Effects of Session Limit and Session Overflow Limit Settings Combinations

Base Session Limit	Session Overflow Limit	Call Handling
0	0	Reject all calls.
10	0	Accept up to 10 sessions.
10	10	Accept up to 20 sessions and mark sessions 11 to 20 as overflow sessions.
0	10	Accept up to 10 sessions and mark sessions 1 to 10 as overflow.
All	0	Accept all calls.
0	All	Accept all calls and mark all calls as overflow.

VPDN Session and Overflow Session Limits

Cisco RPM enables you to configure base and overflow session limits per VPDN profile for managing VPDN sessions.

Note The VDPN session and session overflow limits are independent of the limits set in the customer profiles.

The base VPDN session limit determines the maximum number of nonoverflow sessions supported for a VPDN profile. When the VPDN session limit is reached, if overflow sessions are not enabled, any new VPDN calls using the VPDN profile sessions are rejected. If overflow sessions are enabled, new sessions up to the session overflow limit are processed and marked as overflow for VPDN accounting.

The VPDN session overflow limit determines the number of sessions above the session limit allowed in the VPDN group. If the session overflow limit is greater than zero, overflow sessions are enabled and the maximum number of allowed sessions is the session limit plus the session overflow limit. While the session overflow limit has been reached, any new calls are rejected.

Enabling VPDN overflow sessions is useful for allocating extra sessions for preferred customers at premium rates. Overflow sessions are also useful for encouraging customers to adequately forecast bandwidth usage or for special events when normal session usage is exceeded. For example, if a

customer is having a corporate-wide program and many people are expected to request remote access, you could enable many overflow sessions and charge a premium rate for the extra bandwidth requirements.

VPDN MLP Bundle and Links-per-Bundle Limits

To ensure that resources are not consumed by a few users with MLP connections, Cisco RPM also enables you to specify the maximum number of MLP bundles that can open in a VPDN group. In addition, you can specify the maximum number of links for each MLP bundle.

For example, if standard ISDN users access the VPDN profile, limit this setting to two links per bundle. If video conferencing is used, increase this setting to accommodate the necessary bandwidth (usually six links). These limits have no overflow option and are configured under the VPDN group component.

VPDN Tunnel Limits

For increased VPDN tunnel management, Cisco RPM enables you to set an IP endpoint session limit for each IP endpoint. IP endpoints are configured for VPDN groups.

Figure 105 and Figure 106 show logical flowcharts of RPM call processing for a standalone NAS with and without the RPM Direct Remote Services feature.

Figure 105 RPM Call-Processing Flowchart for a Standalone Network Access Server

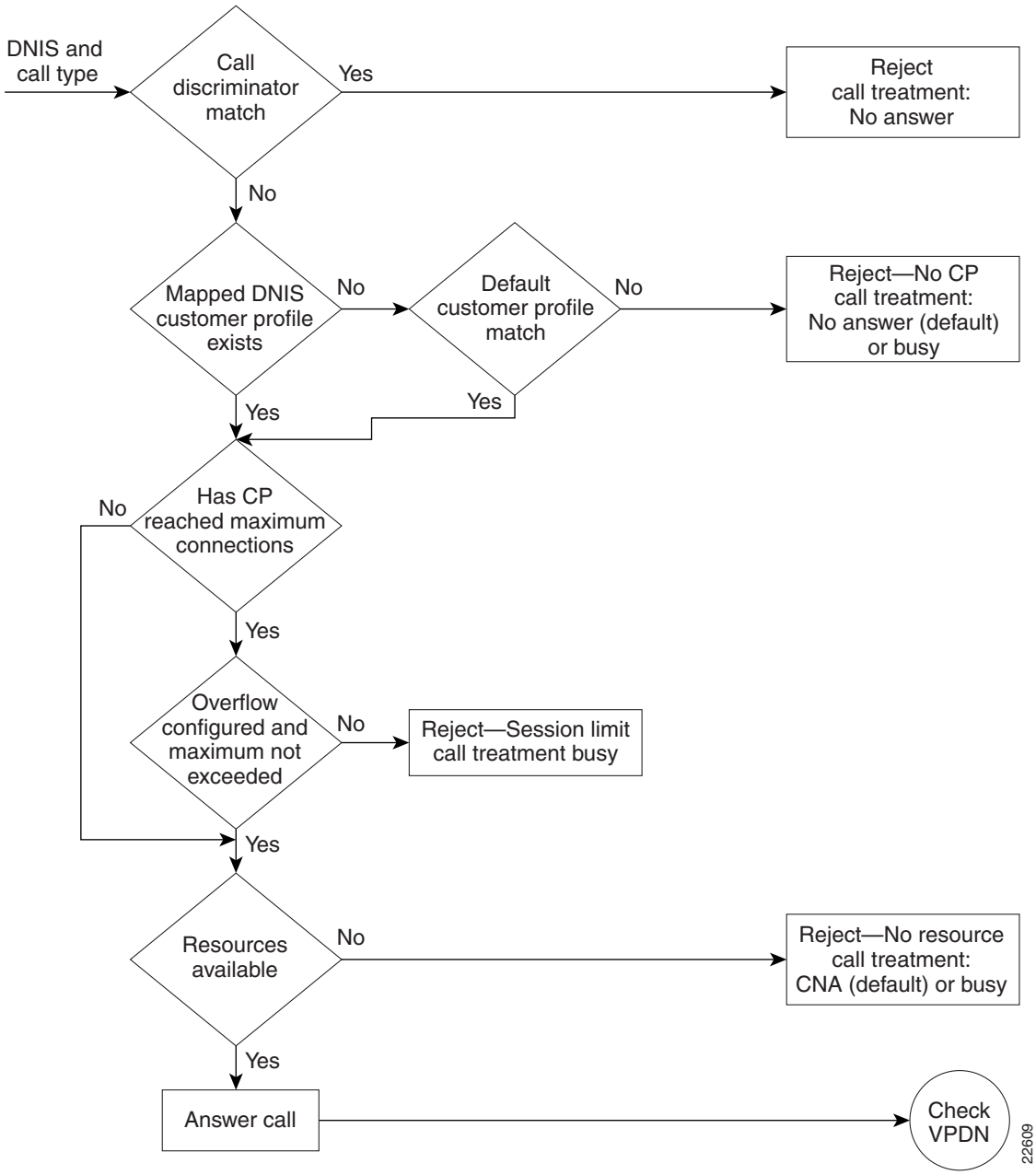
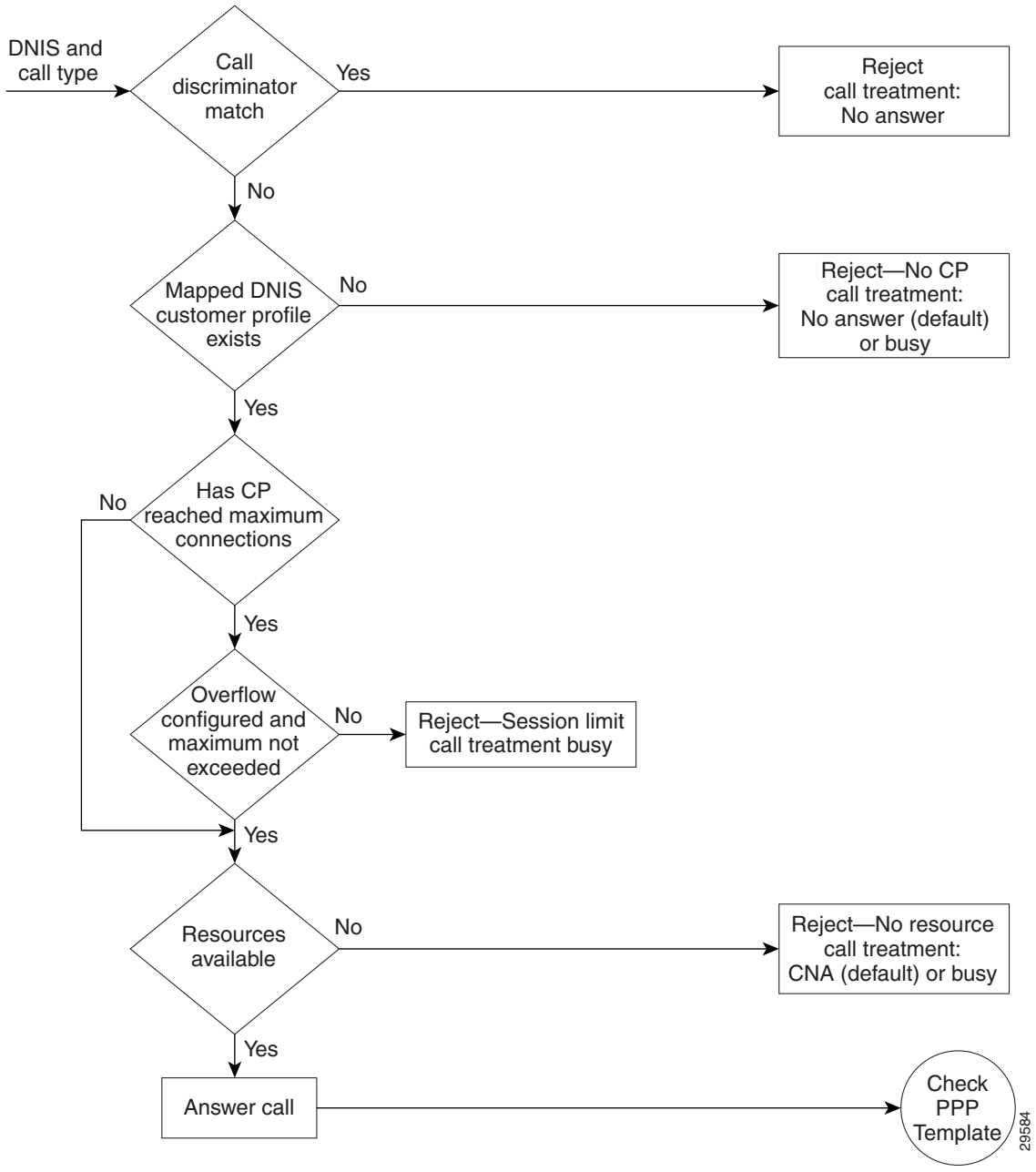


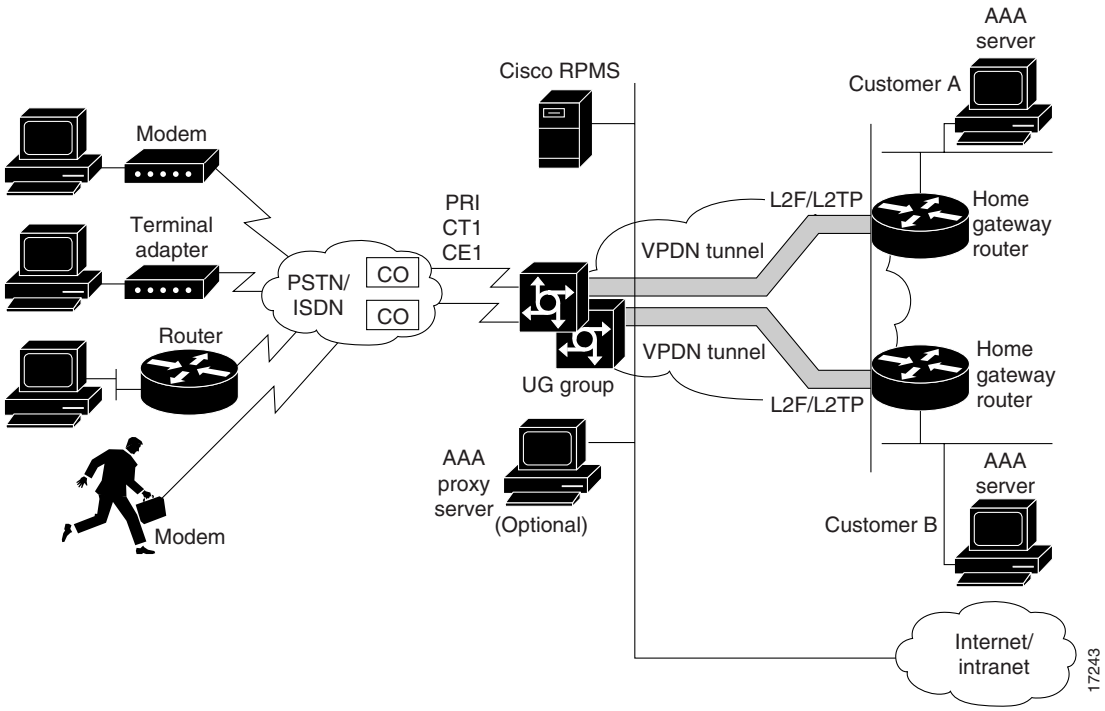
Figure 106 Flowchart for a Standalone Network Access Server with RPM Direct Remote Services



RPM Using the Cisco RPMS

Figure 107 shows a typical resource pooling network scenario using RPMS.

Figure 107 RPM Scenario Using RPMS

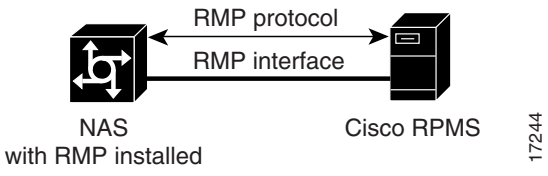


Resource Manager Protocol

Resource Manager Protocol (RMP) is a robust, recoverable protocol used for communication between the Cisco RPMS and the NAS. Each NAS client uses RMP to communicate resource management requests to the Cisco RPMS server. RPMS also periodically polls the NAS clients to query their current call information or address error conditions when they occur. RMP also allows for protocol attributes that make it extensible and enable support for customer billing requirements.

Figure 108 shows the relationship of Cisco RPM CLID/DNIS Call Discriminator Feature and RMP.

Figure 108 Cisco RPM CLID/DNIS Call Discriminator Feature and RMP



Note RMP must be enabled on all NASes that communicate with the Cisco RPM CLID/DNIS Call Discriminator Feature.

Direct Remote Services

Direct remote services is an enhancement to Cisco RPM implemented in Cisco IOS Release 12.0(7)T that enables service providers to implement wholesale dial services without using VPDN tunnels. A customer profile that has been preconfigured with a PPP template to define the unique PPP services for the wholesale dial customer is selected by the incoming DNIS and call type. At the same time, the DNIS is used to select AAA server groups for authentication/authorization and for accounting for the customer.

PPP Common Configuration Architecture (CCA) is the new component of the RPM customer profile that enables direct remote services. The full PPP command set available in Cisco IOS software is configurable per customer profile for wholesale dial applications. A customer profile typically includes the following PPP parameters:

- Local or named IP address pools
- Primary and secondary DNS or WINS addresses
- Authentication method (PAP, CHAP, MS-CHAP)
- Multilink PPP links per bundle limits

The AAA session information is selected by the incoming DNIS. AAA server lists provide the IP addresses of AAA servers for authentication, authorization, and accounting in the wholesale local network of the customer. The server lists for both authentication and authorization and for accounting contain the server addresses, AAA server type, timeout, retransmission, and keys per server.

When direct remote services is implemented on a Cisco NAS, the following sequence occurs:

1. The NAS sends an authorization request packet to the AAA server by using the authentication method (PAP, CHAP, MSCHAP) that has been configured through PPP.
2. The AAA server accepts the authorization request and returns one of the following items to the NAS:
   - A specific IP address
   - An IP address pool name
   - Nothing
3. Depending on the response from the AAA server, the NAS assigns one of the following items to the user through the DNS/WINS:
   - The IP address returned by the AAA server
   - An IP address randomly assigned from the named IP address pool
   - An IP address from a pool specified in the customer profile template

Note If the AAA server sends back to the NAS a named IP address pool and that name does not exist on the NAS, the request for service is denied. If the AAA server does not send anything back to the NAS and there is an IP address pool name configured in the customer profile template, an address from that pool is used for the session.

RPM Process with RPMS and SS7

For information on SS7 implementation for RPM, refer to the document *Cisco Resource Pool Manager Server 1.0 SS7 Implementation.*

Additional Information About Cisco RPM

For more information about Cisco RPM, see the following documents:

- *AAA Server Group*
- *Cisco Access VPN Solutions Using Tunneling Technology*
- *Cisco AS5200 Universal Access Server Software Configuration Guide*
- *Cisco AS5300 Software Configuration Guide*
- *Cisco AS5800 Access Server Software ICG*
- *Cisco Resource Pool Manager Server Configuration Guide*
- *Cisco Resource Pool Manager Server Installation Guide*
- *Cisco Resource Pool Manager Server Solutions Guide*
- *Dial Solutions Quick Configuration Guide*
- *RADIUS Multiple UDP Ports Support*
- *Redundant Link Manager*
- *Release Notes for Cisco Resource Pool Manager Server* Release 1.0
- *Resource Pool Management*
- *Resource Pool Management with Direct Remote Services*
- *Resource Pool Manager Customer Profile Template*
- *Selecting AAA Server Groups Based on DNIS*
- *SS7 Continuity Testing for Network Access Servers*
- *SS7 Dial Solution System Integration*

How to Configure RPM

Read and comply with the following restrictions and prerequisites before beginning RPM configuration:

- RPM is supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 Universal Access Servers
- Modem pooling and RPM are not compatible.
- The Cisco RPM CLID/DNIS Call Discriminator Feature must have Cisco RPM configured.
- CLID screening is not available to channel-associated signaling (CAS) interrupt level calls.
- Cisco RPM requires the NPE 300 processor when implemented on the Cisco AS5800.
- For Cisco AS5200 and Cisco AS5300 access servers, Cisco IOS Release 12.0(4)XI1 or later releases must be running on the NAS.
- For Cisco AS5800, Cisco IOS Release 12.0(5)T or later releases must be running on the NAS.
- A minimum of 64 MB must be available on the DMM cards.
- The RPM application requires an NPE 300.
- For call discriminator profiles, the Cisco AS5300, Cisco AS5400, or Cisco AS5800 Universal Access Servers require a minimum of 16 MB Flash memory and 128 MB DRAM memory, and need to be configured for VoIP as an H.323-compliant gateway.

The following tasks must be performed before configuring RPM:

- Accomplish initial configuration as described in the appropriate *Universal Access Server Software Configuration Guide*. Perform the following tasks as required.
  - Set your local AAA
  - Define your TACACS+ server for RPM
  - Define AAA accounting
  - Ensure PPP connectivity
  - Ensure VPDN connectivity

  Refer to the document *Configuring the NAS for Basic Dial Access* for more information.

To configure your NAS for RPM, perform the following tasks:

- Enabling RPM (Required)
- Configuring DNIS Groups (As required)
- Creating CLID Groups (As required)
- Configuring Discriminator Profiles (As required)
- Configuring Resource Groups (As required)
- Configuring Service Profiles (As required)
- Configuring Customer Profiles (As required)
- Configuring a Customer Profile Template (As required)
- Placing the Template in the Customer Profile (As required)
- Configuring AAA Server Groups (As required)
- Configuring VPDN Profiles (As required)
- Configuring VPDN Groups (As required)
- Counting VPDN Sessions by Using VPDN Profiles (As required)
- Limiting the Number of MLP Bundles in VPDN Groups (As required)
- Configuring Switched 56 over CT1 and RBS (As required)

See the section "Troubleshooting RPM" later in this chapter for troubleshooting tips. See the section "Configuration Examples for RPM" at the end of this chapter for examples of how to configure RPM in your network.

Enabling RPM

To enable RPM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool enable`	Turns on RPM.
Step 2	`Router(config)# resource-pool call treatment resource channel-not-available`	Creates a resource group for resource management.
Step 3	`Router(config)# resource-pool call treatment profile no-answer`	Sets up the signal sent back to the telco switch in response to incoming calls.
Step 4	`Router(config) # resource-pool aaa protocol local`	Specifies which protocol to use for resource management.

Note If you have an RPMS, you need not define VPDN groups/profiles, customer profiles, or DNIS groups on the NAS; you need only define resource groups. Configure the remaining items by using the RPMS system.

Configuring DNIS Groups

This configuration task is optional.

To configure DNIS groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dialer dnis group` *dnis-group-name*	Creates a DNIS group. The name you specify in this step must match the name entered when configuring the customer profile.
Step 2	`Router(config-called-group)# call-type cas` {**digital** | **speech**}	Statically sets the call-type override for incoming CAS calls.
Step 3	`Router(config-called-group)# number` *number*	Enters DNIS numbers to be used in the customer profile. (Wildcards can be used.)

For default DNIS service, no DNIS group configuration is required. The following characteristics and restrictions apply to DNIS group configuration:

- Each DNIS group/call-type combination can apply to only one customer profile.
- You can use up to four default DNIS groups (one for each call type).
- You must statically configure CAS call types.
- You can use x, X or . as wildcards within each DNIS number.

Creating CLID Groups

You can add multiple CLID groups to a discriminator profile. You can organize CLID numbers for a customer or service type into a CLID group. Add all CLID numbers into one CLID group, or subdivide the CLID numbers using criteria such as call type, geographical location, or division. To create CLID groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dialer clid group clid-group-name`	Creates a CLID group, assigns it a name of up to 23 characters, and enters CLID configuration mode. The CLID group must be the same as the group specified in the customer profile configuration. Refer to the *Resource Pool Management with Direct Remote Services* document for information on configuring customer profiles.
Step 2	`Router(config-clid-group)# number clid-group-number`	Enters CLID configuration mode, and adds a CLID number to the dialer CLID group that is used in the customer profile. The CLID number can have up to 65 characters. You can use **x**, **X** or **.** as wildcards within each CLID number. The CLID screening feature rejects this number if it matches the CLID of an incoming call.

Configuring Discriminator Profiles

Discriminator profiles enable you to process calls differently on the basis of the call type and CLID/DNIS combination. The "Call Discriminator Profiles" section earlier in this chapter describes the different types of discriminator profiles that you can create.

To configure discriminator profiles for RPM implementation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile discriminator name`	Creates a call discriminator profile and assigns it a name of up to 23 characters.
Step 2	`Router(config-call-d)# call-type {all | digital | speech | v110 | v120}`	Specifies the type of calls you want to block. The NAS will not answer the call-type you specify.

	Command	Purpose
Step 3	`Router(config-call-d)# clid group {clid-group-name | default}`	Optional. Associates a CLID group with the discriminator. If you do not specify a *clid-group-nam*e, the default discriminator in the RM is used. Any CLID number coming in on a call is in its respective default group unless it is specifically assigned a *clid-group-nam*e. After a CLID group is associated with a call type in a discriminator, it cannot be used in any other discriminator.
Step 4	`Router(config-call-d)# dnis group {dnis-group-name | default}`	Optional. Associates a DNIS group with the discriminator. If you do not specify a *dnis-group-nam*e, the default discriminator in the RM is used. Any DNIS number coming in on a call is in its respective default group unless it is specifically assigned a *dnis-group-nam*e. After a DNIS group is associated with a call type in a discriminator, it cannot be used in any other discriminator.

To verify discriminator profile settings, use the following commands:

Step 1 Use the **show resource-pool discriminator** *name* command to verify the call discriminator profiles that you configured.

If you enter the **show resource-pool discriminator** command without including a call discriminator name, a list of all current call discriminator profiles appears.

If you enter a call discriminator profile *name* with the **show resource-pool discriminator** command, the number of calls rejected by the selected call discriminator appears.

```
Router# show resource-pool discriminator

List of Call Discriminator Profiles:
    deny_CLID

Router# show resource-pool discriminator deny_CLID

    1 calls rejected
```

Step 2 Use the **show dialer** command to display general diagnostic information for interfaces configured for the dialer.

```
Router# show dialer [interface] type number
```

Configuring Resource Groups

To configure resource groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool group resource name`	Creates a resource group and assign it a name of up to 23 characters.
Step 2	`Router(config-resource-group)# range {port {slot/port slot/port}} | {limit number}`	Associates a range of modems or other physical resources with this resource group: • For port-based resources, use the physical locations of the resources. • For non-port-based resources, use a single integer limit. Specify the maximum number of simultaneous connections supported by the resource group. Up to 192 connections may be supported, depending on the hardware configuration of the access server.

For external Cisco RPMS environments, configure resource groups on the NAS before defining them on external RPMS servers.

For standalone NAS environments, first configure resource groups before using them in customer profiles.

Resource groups can apply to multiple customer profiles.

Note You can separate physical resources into groups. However, do not put heterogeneous resources in the same group. Do not put MICA technologies modems in the same group as Microcom modems. Do not put modems and HDLC controllers in the same resource group. Do not configure the **port** and **limit** command parameters in the same resource group.

Configuring Service Profiles

To configure service profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile service name`	Creates a service profile and assign it a name of up to 23 characters.
Step 2	`Router(config-service-profil)# modem min-speed {speed | any} max-speed {speed | any [modulation value]}`	Specifies the desired modem parameter values. The range for **min-speed** and **max-speed** is 300 to 56000 bits per second.

Service profiles are used to configure modem service parameters for Nextport and MICA technologies modems, and support speech, digital, V.110, and V.120 call types. Error-correction and compression are hidden parameters that may be included in a service profile.

Configuring Customer Profiles

To configure customer profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Creates a customer profile.
Step 2	`Router(config-customer-pro)# dnis group` {*dnis-group-name* | **default**}	Includes a group of DNIS numbers in the customer profile.
Step 3	`Router(config-customer-pro)# limit base-size` {*number* | **all**}	Specifies the base size usage limit.
Step 4	`Router(config-customer-pro)# limit overflow-size` {*number* | **all**}	Specifies the oversize size usage limit.
Step 5	`Router(config-customer-pro)# resource` *WORD* {**digital** | **speech** | **v110** | **v120**} [**service** *WORD*]	Assigns resources and supported call types to the customer profile.

Customer profiles are used so that service providers can assign different service characteristics to different customers. Note the following characteristics of customer profiles:

- Multiple resources of the same call type are used sequentially.
- The limits imposed are per customer (DNIS)—not per resource.
- A digital resource with a call type of **speech** allows for Data over Speech Bearer Service (DoSBS).

Configuring Default Customer Profiles

Default customer profiles are identical to standard customer profiles, except they do not have any associated DNIS groups. To define a default customer profile, use the reserved keyword **default** for the DNIS group:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to the default customer profile.
Step 2	`Router(config-customer-pro)# dnis group default`	Assigns the default DNIS group to the customer profile. This sets up the customer profile such that it will use the default DNIS configuration, which is automatically set on the NAS.

The rest of the customer profile is configured as shown in the previous section "Configuring Customer Profiles."

Configuring Customer Profiles Using Backup Customer Profiles

Backup customer profiles are customer profiles configured locally on the Cisco NAS and are used to answer calls on the basis of a configured allocation scheme when the link between the Cisco NAS and Cisco RPMS is disabled.

To enable the backup feature, you need to have already configured the following on the router:

- The **resource-pool aaa protocol group** *name* **local** command.
- All customer profiles and DNIS groups on the NAS.

The backup customer profile can contain all of the elements defined in a standard customer profile, including base size or overflow parameters. However, when the connection between the Cisco NAS and Cisco RPMS is unavailable, session counting and session limits are not applied to incoming calls. Also, after the connection is reestablished, there is no synchronization of call counters between the Cisco NAS and Cisco RPMS.

Configuring Customer Profiles for Using DoVBS

To configure customer profiles for using DoVBS, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to a customer profile.
Step 2	`Router(config-customer-pro)# dnis group` *name*	Assigns a DNIS group to the customer profile. DNIS numbers are assigned as shown in the previous section.
Step 3	`Router(config)# limit base-size` {*number* | **all**}	Specifies the VPDN base size usage limit.
Step 4	`Router(config)# limit overflow-size` {*number* | **all**}	Specifies the VPDN overflow size usage limit.
Step 5	`Router(config-customer-pro)# resource` *name* {**digital** | **speech** | **v110** | **v120**} [**service** *name*]	Specifies resource names to use within the customer profile.

To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource. The DNIS group assigned to the customer profile should have a call type of speech. The resource group assigned to this customer profile will be digital resources and also have a call type of speech, so the call will terminate on an HDLC controller rather than a modem.

See the section "Customer Profile Configuration for DoVBS Example" at the end of this chapter for a configuration example.

Configuring a Customer Profile Template

Customer profile templates provide a way to keep each unique situation for a customer separate for both security and accountability. This is an optional configuration task.

To configure a template and place it in a customer profile, ensure that all basic configuration tasks and the RPM configuration tasks have been completed and verified before attempting to configure the customer profile templates.

To add PPP configurations to a customer profile, create a customer profile template. Once you create the template and associate it with a customer profile by using the **source template** command, it is integrated into the customer profile.

To configure a template in RPM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# template name`	Creates a customer profile template and assign a unique name that relates to the customer that will be receiving it. **Note** Steps 2, 3, and 4 are optional. Enter multilink, peer, and ppp commands appropriate to the application requirements of the customer.
Step 2	`Router(config-template)# peer default ip address pool pool-name`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.
Step 3	`Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 4	`Router(config-template)# ppp multilink`	(Optional) Enables Multilink PPP for this customer profile.
Step 5	`Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.
Step 6	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this template.
Step 7	`Router(config-customer-profi)# source template name`	Attaches the customer profile template you have just configured to the customer profile.

Typical Template Configuration

The following example shows a typical template configuration:

```
template Word
  multilink {max-fragments frag-num | max-links num | min-links num}
  peer match aaa-pools
  peer default ip address {pool pool-name1 [pool-name2] | dhcp}
  ppp ipcp {dns | wins} A.B.C.D [W.X.Y.Z]
resource-pool profile customer WORD
  source template Word
  aaa group-configuration aaa-group-name

template acme_direct
  peer default ip address pool tahoe
  ppp authentication chap isdn-users
  ppp multilink
```

Verifying Template Configuration

To verify your template configuration, perform the following steps:

Step 1 Enter the **show running-config** EXEC command (where the template name is “PPP1”):

```
Router#
Router# show running-config begin template
.
.
.
```

```
template PPP1
peer default ip address pool pool1 pool2
ppp ipcp dns 10.1.1.1 10.1.1.2
ppp ipcp wins 10.1.1.3 10.1.1.4
ppp multilink max-links 2
.
.
.
```

Step 2 Ensure that your template appears in the configuration file.

Placing the Template in the Customer Profile

To place your template in the customer profile, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to a customer profile.
Step 2	`Router(config-customer-pr)# source template`	Associates the template with the customer profile.

To verify the placement of your template in the customer profile, perform the following steps:

Step 1 Enter the **show resource-pool customer** EXEC command:

```
Router# show resource-pool customer

List of Customer Profiles:
    CP1
    CP2
```

Step 2 Look at the list of customer profiles and make sure that your profile appears in the list.

Step 3 To verify a particular customer profile configuration, enter the **show resource-pool customer** *name* EXEC command (where the customer profile name is “CP1”):

```
Router# show resource-pool customer CP1

97 active connections
    120 calls accepted
    210 max number of simultaneous connections
    50 calls rejected due to profile limits
    0 calls rejected due to resource unavailable
    90 minutes spent with max connections
    5 overflow connections
    2 overflow states entered
    0 overflow connections rejected
    0 minutes spent in overflow
    13134 minutes since last clear command
```

Configuring AAA Server Groups

To configure AAA server groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA on the NAS.

Step 2	`Router(config)# radius-server key key`
	or
	`Router(config)# tacacs-server key key`
		Set the authentication and encryption key used for all RADIUS or TACACS+ communications between the NAS and the RADIUS or TACACS+ daemon.

Step 3	`Router(config)# radius-server host {hostname | ip-address key} [auth-port port acct-port port]`
	or
	`Router(config)# tacacs-server host ip-address key`
		Specifies the host name or IP address of the server host before configuring the AAA server group. You can also specify the UDP destination ports for authentication and for accounting.

Step 4	`Router(config)# aaa group server {radius | tacacs+} group-name`	Selects the AAA server type you want to place into a server group and assign a server group name.
Step 5	`Router(config-sg radius)# server ip-address`	Specifies the IP address of the selected server type. This must be the same IP address that was assigned to the server host in Step 3.
Step 6	`Router(config-sg radius)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this AAA server group.
Step 8	`Router(config-customer-profil)# aaa group-configuration group-name`	Associates this AAA server group (named in Step 4) with the customer profile named in Step 7.

AAA server groups are lists of AAA server hosts of a particular type. The Cisco RPM currently supports RADIUS and TACACS+ server hosts. A AAA server group lists the IP addresses of the selected server hosts.

You can use a AAA server group to define a distinct list of AAA server hosts and apply this list to the Cisco RPM application. Note that the AAA server group feature works only when the server hosts in a group are of the same type.

Configuring VPDN Profiles

A VPDN profile is required only if you want to impose limits on the VPDN tunnel that are separate from the customer limits.

To configure VPDN profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile vpdn profile-name`	Creates a VPDN profile and assigns it a profile name
Step 2	`Router(config-vpdn-profile)# limit base-size {number | all}`	Specifies the maximum number of simultaneous base VPDN sessions to be allowed for this VPDN group under the terms of the service-level agreement (SLA). The range is 0 to 1000 sessions. If all sessions are to be designated as base VPDN sessions, specify **all**.
Step 3	`Router(config-vpdn-profile)# limit overflow-size {number | all}`	Specifies the maximum number of simultaneous overflow VPDN sessions to be allowed for this VPDN group under the terms of the SLA. The range is 0 to 1000 sessions. If all sessions are to be designated as overflow VPDN sessions, specify **all**.
Step 4	`Router(config-vpdn-profile)# exit`	Returns to global configuration mode.
Step 5	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this VPDN group.

Step 6	`Router(config-customer-profi)# vpdn profile profile-name`
	or
	`Router(config-customer-profi)# vpdn group group-name`
		Attaches the VPDN profile you have just configured to the customer profile to which it belongs, or, if the limits imposed by the VPDN profile are not required, attaches VPDN group instead (see the section "Configuring VPDN Groups" later in this chapter).

Configuring VPDN Groups

To configure VPDN groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN sessions on the NAS.
Step 2	`Router(config)# vpdn-group group-name`	Creates a VPDN group and assigns it a unique name. Each VPDN group can have multiple endpoints (HGW/LNSs).
Step 3	`Router(config-vpdn)# request dialin {l2f | l2tp} {ip ip-address} {domain domain-name | dnis dnis-number}`	Specifies the tunneling protocol to be used to reach the remote peer defined by a specific IP address if a dial-in request is received for the specified domain name or DNIS number. The IP address that qualifies the session is automatically generated and need not be entered again.
Step 4	`Router(config-vpdn)# multilink {bundle-number | link-number}`	Specifies the maximum number of bundles and links for all multilink users in the VPDN group. The range for both bundles and links is 0 to 32767. In general, each user requires one bundle.

	Command	Purpose
Step 5	`Router(config-vpdn)# loadsharing ip ip-address [limit number]`	Configures the endpoints for loadsharing. This router will share the load of IP traffic with the first router specified in Step 2. The **limit** keyword limits the number of simultaneous sessions that are sent to the remote endpoint (HGW/LNS). This limit can be 0 to 32767 sessions.
Step 6	`Router(config-vpdn)# backup ip ip-address [limit number] [priority number]`	Sets up a backup HGW/LNS router. The number of sessions per backup can be limited. The priority number can be 2 to 32767. The highest priority is 2, which is the first HGW/LNS router to receive backup traffic. The lowest priority, which is the default, is 32767.
Step 7	`Router(config-vpdn)# exit`	Returns to global configuration mode.

Step 8	`Router(config)# resource-pool profile vpdn profile-name`
	or
	`Router(config)# resource-pool profile customer name`
		Enters either VPDN profile configuration mode or customer profile configuration mode, depending on whether you want to allow VPDN connections for a customer profile, or allow combined session counting on all of the VPDN sessions within a VPDN profile.

Step 9	`Router(config-vpdn-profile)# vpdn group group-name`
	or
	`Router(config-customer-profi)# vpdn group group-name`
		Attaches the VPDN group to either the VPDN profile or the customer profile specified in Step 8.

A VPDN group consists of VPDN sessions that are combined and placed into a customer profile or a VPDN profile. Note the following characteristics of VPDN groups:

- The *dnis-group-name* argument is required to authorize the VPDN group with RPM.
- A VPDN group placed in a customer profile allows VPDN connections for the customer using that profile.
- A VPDN group placed in a VPDN profile allows the session limits configured for that profile to apply to all of the VPDN sessions within that VPDN group.
- VPDN data includes an associated domain name or DNIS, an endpoint IP address, the maximum number of MLP bundles, and the maximum number of links per MLP bundle; this data can optionally be located on a AAA server.

See the sections "VPDN Configuration Example" and "VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example" at the end of this chapter for examples of using VPDN with RPM.

Counting VPDN Sessions by Using VPDN Profiles

Session counting is provided for each VPDN profile. One session is brought up each time a remote client dials into a HGW/LNS router by using the NAS/LAC. Sessions are counted by using VPDN profiles. If you do not want to count the number of VPDN sessions, do not set up any VPDN profiles. VPDN profiles count sessions in one or more VPDN groups.

To configure VPDN profile session counting, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile vpdn` *name*	Creates a VPDN profile.
Step 2	`Router(config-vpdn-profile)# vpdn-group` *name* `Router(config-vpdn-profile)# exit`	Associates a VPDN group to the VPDN profile. VPDN sessions done within this VPDN group will be counted by the VPDN profile.
Step 3	`Router(config)# resource-pool profile customer` *name* `Router(config-customer-profi)# vpdn profile` *name*	Links the VPDN group to a customer profile.
Step 4	`Router(config-customer-profi)# ^Z` `Router#`	Returns to EXEC mode to perform verification steps.

To verify session counting and view VPDN group information configured under resource pooling, use the **show resource-pool vpdn group** command. In this example, two different VPDN groups are configured under two different customer profiles:

```
Router# show resource-pool vpdn group

List of VPDN Groups under Customer Profiles
Customer Profile customer1:customer1-vpdng
Customer Profile customer2:customer2-vpdng
List of VPDN Groups under VPDN Profiles
VPDN Profile customer1-profile:customer1-vpdng
```

To display the contents of a specific VPDN group, use the **show resource-pool vpdn group** *name* command. This example contains one domain name, two DNIS called groups, and two endpoints:

```
Router# show resource-pool vpdn group customer2-vpdng

VPDN Group customer2-vpdng found under Customer Profiles: customer2

Tunnel (L2TP)
------
dnis:cg1
dnis:cg2
dnis:jan

Endpoint        Session Limit Priority Active Sessions Status Reserved Sessions
--------        ------------- -------- --------------- ------ -----------------
172.21.9.67     *             1        0               OK     -
10.1.1.1         *             2        0               OK     -
--------------- -------------           ---------------        -----------------
Total           *                      0                      0
```

To display the contents of a specific VPDN profile, use the **show resource-pool vpdn profile** *name* command, as follows:

```
Router# show resource-pool vpdn profile ?

  WORD  VPDN profile name
  <cr>

Router# show resource-pool vpdn profile customer1-profile

0 active connections
0 max number of simultaneous connections
0 calls rejected due to profile limits
```

```
0 calls rejected due to resource unavailable
0 overflow connections
0 overflow states entered
0 overflow connections rejected
1435 minutes since last clear command
```

Note Use the **debug vpdn event** command to troubleshoot VPDN profile limits, session limits, and MLP connections. First, enable this command; then, send a call into the access server. Interpret the debug output and make configuration changes as needed.

To debug the L2F or L2TP protocols, use the **debug vpdn l2x** command:

```
Router# debug vpdn l2x ?

  error            VPDN Protocol errors
  event            VPDN event
  l2tp-sequencing  L2TP sequencing
  l2x-data         L2F/L2TP data packets
  l2x-errors       L2F/L2TP protocol errors
  l2x-events       L2F/L2TP protocol events
  l2x-packets      L2F/L2TP control packets
  packet           VPDN packet
```

Limiting the Number of MLP Bundles in VPDN Groups

Cisco IOS software enables you to limit the number of MLP bundles and links supported for each VPDN group. A bundle name consists of a username endpoint discriminator (for example, an IP address or phone number) sent during LCP negotiation.

To limit the number of MLP bundles in VPDN groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group` *name*	Creates a VPDN group.
Step 2	`Router(config-vpdn)# multilink` {**bundle** *number* | **link** *number*}	Limits the number of MLP bundles per VPDN group and links per bundle.[1] These settings limit the number of users that can multilink.

1. Both the NAS/LAC and the HGW/LNS router must be configured to support multilink before a client can use multilink to connect to a HGW/LNS.

The following example shows the **show vpdn multilink** command output for verifying MLP bundle limits:

```
Router# show vpdn multilink

Multilink Bundle Name   VPDN Group Active links Reserved links Bundle/Link Limit
---------------------   ---------- ------------ -------------- -----------------
twv@anycompany.com      vgdnis     0            0              */*
```

Note Use the **debug vpdn event** and **debug resource-pooling** commands to troubleshoot VPDN profile limits, session limits, and MLP connections. First, enable this command; then, send a call into the access server. Interpret the debug output and make configuration changes as needed.

Configuring Switched 56 over CT1 and RBS

To configure switched 56 over CT1 and RBS, use the following commands beginning in global configuration mode. Perform this task on the Cisco AS5200 and Cisco AS5300 access servers only.

	Command	Purpose
Step 1	`Router(config)# controller t1 number`	Specifies a controller and begins controller configuration mode.
Step 2	`Router(config-controller)# cas-group 0 timeslots 1-24 type e&m-fgb {dtmf | mf} {dnis}`	Creates a CAS group and assigns time slots.
Step 3	`Router(config-controller)# framing {sf | esf}`	Specifies framing.
Step 4	`Router(config-controller)# linecode {ami | b8zs}`	Enters the line code.
Step 5	`Router(config-controller)# exit`	Returns to global configuration mode.
Step 6	`Router(config)# dialer dnis group name`	Creates a dialer called group.
Step 7	`Router(config-called-group)# call-type cas digital`	Assigns a call type as digital (switch 56).
Step 8	`Router(config-called-group)# exit`	Returns to global configuration mode.

Step 9	`Router(config)# interface serial number:number`
	
	`Router(config-if)#`
		Specifies the logical serial interface, which was dynamically created when the **cas-group** command was issued.
		
		This command also enters interface configuration mode, where you configure the core protocol characteristics for the serial interface.

To verify switched 56 over CT1, use the **show dialer dnis** command as follows:

```
Router# show dialer dnis group

List of DNIS Groups:
  default
  mdm_grp1

Router# show dialer dnis group mdm_grp1

Called Number:2001
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2002
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2003
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2004
  0 total connections
  0 peak connections
  0 calltype mismatches
.
.
.
```

```
Router# show dialer dnis number

List of Numbers:
  default
  2001
  2002
  2003
  2004
.
.
.
```

Verifying RPM Components

The following sections provide call-counter and call-detail output for the different RPM components:

- Verifying Current Calls
- Verifying Call Counters for a Customer Profile
- Clearing Call Counters
- Verifying Call Counters for a Discriminator Profile
- Verifying Call Counters for a Resource Group
- Verifying Call Counters for a DNIS Group
- Verifying Call Counters for a VPDN Profile
- Verifying Load Sharing and Backup

Verifying Current Calls

The following output from the **show resource-pool call** command shows the details for all current calls, including the customer profile and resource group, and the matched DNIS group:

```
Router# show resource-pool call

Shelf 0, slot 0, port 0, channel 15, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group isdn-ports
  DNIS number 301001
Shelf 0, slot 0, port 0, channel 14, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group isdn-ports
  DNIS number 301001
Shelf 0, slot 0, port 0, channel 11, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group MICA-modems
  DNIS number 301001
```

Verifying Call Counters for a Customer Profile

The following output from the **show resource-pool customer** command shows the call counters for a given customer profile. These counters include historical data and can be cleared.

```
Router# show resource-pool customer ACME

    3 active connections
    41 calls accepted
    3 max number of simultaneous connections
```

```
    11 calls rejected due to profile limits
    2 calls rejected due to resource unavailable
    0 minutes spent with max connections
    5 overflow connections
    1 overflow states entered
    11 overflow connections rejected
    10 minutes spent in overflow
    214 minutes since last clear command
```

Clearing Call Counters

The **clear resource-pool** command clears the call counters.

Verifying Call Counters for a Discriminator Profile

The following output from the **show resource-pool discriminator** command shows the call counters for a given discriminator profile. These counters include historical data and can be cleared.

```
Router# show resource-pool discriminator

List of Call Discriminator Profiles:
    deny_DNIS

Router# show resource-pool discriminator deny_DNIS

    1 calls rejected
```

Verifying Call Counters for a Resource Group

The following output from the **show resource-pool resource** command shows the call counters for a given resource group. These counters include historical data and can be cleared.

```
Router# show resource-pool resource

List of Resources:
    isdn-ports
    MICA-modems

Router# show resource-pool resource isdn-ports

    46 resources in the resource group
    2 resources currently active
    8 calls accepted in the resource group
    2 calls rejected due to resource unavailable
    0 calls rejected due to resource allocation errors
```

Verifying Call Counters for a DNIS Group

The following output from the **show dialer dnis** command shows the call counters for a given DNIS group. These counters include historical data and can be cleared.

```
Router# show dialer dnis group ACME_dnis_numbers

DNIS Number:301001
  11 total connections
  5 peak connections
  0 calltype mismatches
```

Verifying Call Counters for a VPDN Profile

The following output from the **show resource-pool vpdn** command shows the call counters for a given VPDN profile or the tunnel information for a given VPDN group. These counters include historical data and can be cleared.

```
Router# show resource-pool vpdn profile ACME_VPDN

        2 active connections
        2 max number of simultaneous connections
        0 calls rejected due to profile limits
        0 calls rejected due to resource unavailable
        0 overflow connections
        0 overflow states entered
        0 overflow connections rejected
        215 minutes since last clear command

Router# show resource-pool vpdn group outgoing-2

VPDN Group outgoing-2 found under VPDN Profiles:  ACME_VPDN

Tunnel (L2F)
------
dnis:301001
dnis:ACME_dnis_numbers

Endpoint    Session Limit Priority Active Sessions Status Reserved Sessions
--------    ------------- -------- --------------- ------ -----------------
172.16.1.9  *             1        2               OK     -
--------    -------------          ---------------        -----------------
Total       *                      2                      0
```

Verifying Load Sharing and Backup

The following example from the **show running-config** EXEC command shows two different VPDN customer groups:

```
Router# show running-config

Building configuration...
.
.
.
vpdn-group customer1-vpdng
 request dialin
 protocol l2f
 domain cisco.com
```

```
 domain cisco2.com
 dnis customer1-calledg
 initiate-to ip 172.21.9.67
 loadsharing ip 172.21.9.68 limit 100
 backup ip 172.21.9.69 priority 5
vpdn-group customer2-vpdng
 request dialin
 protocol l2tp
 dnis customer2-calledg
 domain acme.com
 initiate-to ip 172.22.9.5
```

Troubleshooting RPM

Test and verify that ISDN, CAS, SS7, PPP, AAA, and VPDN are working properly before implementing RPM. Once RPM is implemented, the only **debug** commands needed for troubleshooting RPM are as follows:

- **debug resource pool**
- **debug aaa authorization**

The **debug resource-pool** command is useful as a first step to ensure proper operation. It is usually sufficient for most cases. Use the **debug aaa authorization** command for troubleshooting VPDN and modem service problems.

Problems that might typically occur are as follows:

- No DNIS group found or no customer profile uses a default DNIS
- Call discriminator blocks the DNIS
- Customer profile limits exceeded
- Resource group limits exceeded

Note Always enable the debug and log time stamps when troubleshooting RPM.

This section provides the following topics for troubleshooting RPM:

Resource-Pool Component

The resource-pool component contains two modules—a dispatcher and a local resource-pool manager. The dispatcher interfaces with the signaling stack, resource-group manager, and AAA, and is responsible for maintaining resource-pool call state and status information. The state transitions can be displayed by enabling the resource-pool debug traces. Table 45 summarizes the resource pooling states.

Table 45 Resource Pooling States

State	Description
RM_IDLE	No call activity.
RM_RES_AUTHOR	Call waiting for authorization; message sent to AAA.
RM_RES_ALLOCATING	Call authorized; resource group manager allocating.
RM_RES_ALLOCATED	Resource allocated; connection acknowledgment sent to signaling state. Call should get connected and become active.
RM_AUTH_REQ_IDLE	Signaling module disconnected call while in RM_RES_AUTHOR. Waiting for authorization response from AAA.
RM_RES_REQ_IDLE	Signaling module disconnected call while in RM_RES_ALLOCATING. Waiting for resource allocation response from resource group manager.

The resource-pool state can be used to isolate problems. For example, if a call fails authorization in the RM_RES_AUTHOR state, investigate further with AAA authorization debugs to determine whether the problem lies in the resource-pool manager, AAA, or dispatcher.

The resource-pool component also contains local customer profiles and discriminators, and is responsible for matching, configuring, and maintaining the associated counters and statistics. The resource-pool component is responsible for the following:

- Configuration of customer profiles or discriminators
- Matching a customer profile or discriminator for local profile configuration
- Counters/statistics for customer profiles or discriminators
- Active call information displayed by the **show resource-pool call** command

The RPMS debug commands are summarized in Table 46.

Table 46 Debug Commands for RPM

Command	Purpose
`debug resource-pool`	This debug output should be sufficient for most RPM troubleshooting situations.
`debug aaa authorization`	This debug output provides more specific information and shows the actual DNIS numbers passed and call types used.

Successful Resource Pool Connection

The following sample output from the **debug resource-pool** command displays a successful RPM connection. The entries in bold are of particular importance.

```
*Mar  1 02:14:57.439: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:21
*Mar  1 02:14:57.439: RM: event incoming call
*Mar  1 02:14:57.443: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:21
*Mar  1 02:14:57.447: RM:RPM event incoming call
*Mar  1 02:14:57.459: RPM profile ACME found
*Mar  1 02:14:57.487: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_SUCCESS
 DS0:0:0:0:21
*Mar  1 02:14:57.487: Allocated resource from res_group isdn-ports
*Mar  1 02:14:57.491: RM:RPM profile "ACME", allocated resource "isdn-ports" successfully
*Mar  1 02:14:57.495: RM state:RM_RPM_RES_ALLOCATING event:RM_RPM_RES_ALLOC_SUCCESS
DS0:0:0:0:21
*Mar  1 02:14:57.603: %LINK-3-UPDOWN: Interface Serial0:21, changed state to up
*Mar  1 02:15:00.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:21, changed
state to up
```

Dialer Component

The dialer component contains DNIS groups and is responsible for configuration, and maintenance of counters and statistics. The resource-pool component is responsible for the following:

- DNIS number statistics or counters
- Configuring DNIS groups

Resource Group Manager

Resource groups are created, maintained, allocated, freed, and tallied by the resource group manager. The resource group manager is also responsible for service profiles, which are applied to resources at call setup time. The resource group manager is responsible for:

- Allocating resources when the profile has been authorized and a valid resource group is received
- Statistics or configuration of resource groups
- Configuring or applying service profiles to resource groups
- Collecting DNIS number information for channel-associated signaling calls

Signaling Stack

The signaling stacks currently supported in resource pooling are CAS and ISDN. The signaling stack delivers the incoming call to the resource-pool dispatcher and provides call-type and DNIS number information to the resource-pool dispatcher. Depending on configuration, call connect attempts may fail if the signaling stacks do not send the DNIS number and the call type to the resource-pool dispatcher. Call attempts will also fail if signaling stacks disconnect prematurely, not giving enough time for authorization or resource allocation processes to complete.

Therefore, investigate the signaling stack when call attempts or call treatment behavior does not meet expectations. For ISDN, the **debug isdn q931** command can be used to isolate errors between resource pooling, signaling stack, and switch. For CAS, the **debug modem csm**, **service internal**, and

modem-mgmt csm debug-rbs commands are used on Cisco AS5200 and Cisco AS5300 access servers, while the **debug csm** and **debug trunk cas port** *number* **timeslots** *number* commands are used on the Cisco AS5800 access server.

AAA Component

In context with resource pooling, the AAA component is responsible for the following:

- Authorization of profiles between the resource-pool dispatcher and local or external resource-pool manager
- Accounting messages between the resource-pool dispatcher and external resource-pool manager for resource allocation
- VPDN authorization between VPDN and the local or external resource-pool manager
- VPDN accounting messages between VPDN and the external resource-pool manager
- Overflow accounting records between the AAA server and resource-pool dispatcher
- Resource connect speed accounting records between the AAA server and resource group

VPDN Component

The VPDN component is responsible for the following:

- Creating VPDN groups and profiles
- Searching or matching groups based on domain or DNIS
- Maintaining counts and statistics for the groups and profiles
- Setting up the tunnel between the NAS/LAC and HGW/LNS

The VPDN component interfaces with AAA to get VPDN tunnel authorization on the local or remote resource-pool manager. VPDN and AAA debugging traces should be used for troubleshooting.

Troubleshooting DNIS Group Problems

The following output from the **debug resource-pool** command displays a customer profile that is not found for a particular DNIS group:

```
*Mar  1 00:38:21.011: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:3
*Mar  1 00:38:21.011: RM: event incoming call
*Mar  1 00:38:21.015: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:3
*Mar  1 00:38:21.019: RM:RPM event incoming call
*Mar  1 00:38:21.103: RPM no profile found for call-type digital in default DNIS number
*Mar  1 00:38:21.155: RM:RPM profile rejected do not allocate resource
*Mar  1 00:38:21.155: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:3
*Mar  1 00:38:21.163: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:3
```

Troubleshooting Call Discriminator Problems

The following output from the **debug resource-pool** command displays an incoming call that is matched against a call discriminator profile:

```
*Mar  1 00:35:25.995: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:4
*Mar  1 00:35:25.999: RM: event incoming call
*Mar  1 00:35:25.999: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:4
*Mar  1 00:35:26.003: RM:RPM event incoming call
*Mar  1 00:35:26.135: RM:RPM profile rejected do not allocate resource
*Mar  1 00:35:26.139: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:4
*Mar  1 00:35:26.143: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:4
```

Troubleshooting Customer Profile Counts

The following output from the **debug resource-pool** command displays what happens once the customer profile limits have been reached:

```
*Mar  1 00:43:33.275: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:9
*Mar  1 00:43:33.279: RM: event incoming call
*Mar  1 00:43:33.279: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:9
*Mar  1 00:43:33.283: RM:RPM event incoming call
*Mar  1 00:43:33.295: RPM count exceeded in profile ACME
*Mar  1 00:43:33.315: RM:RPM profile rejected do not allocate resource
*Mar  1 00:43:33.315: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:9
*Mar  1 00:43:33.323: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:9
```

Troubleshooting Resource Group Counts

The following output from the **debug resource-pool** command displays the resources within a resource group all in use:

```
*Mar  1 00:52:34.411: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:19
*Mar  1 00:52:34.411: RM: event incoming call
*Mar  1 00:52:34.415: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:19
*Mar  1 00:52:34.419: RM:RPM event incoming call
*Mar  1 00:52:34.431: RPM profile ACME found
*Mar  1 00:52:34.455: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_SUCCESS
DS0:0:0:0:19
*Mar  1 00:52:34.459: All resources in res_group isdn-ports are in use
*Mar  1 00:52:34.463: RM state:RM_RPM_RES_ALLOCATING event:RM_RPM_RES_ALLOC_FAIL
DS0:0:0:0:19
*Mar  1 00:52:34.467: RM:RPM failed to allocate resources for "ACME"
```

Troubleshooting VPDN

Troubleshooting problems that might typically occur are as follows:

- Customer profile is not associated with a VPDN profile or VPDN group (the call will be locally terminated in this case. Regular VPDN can still succeed even if RPM/VPDN fails).
- VPDN profile limits have been reached (call answered but disconnected).
- VPDN group limits have been reached (call answered but disconnected).
- VPDN endpoint is not reachable (call answered but disconnected).

Troubleshooting RPM/VPDN Connection

The following sample output from the **debug resource-pool** command displays a successful RPM/VPDN connection. The entries in bold are of particular importance.

```
*Mar  1 00:15:53.639: Se0:10 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 00:15:53.655: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 6/0/0/0
*Mar  1 00:15:53.659: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 00:15:53.695: Se0:10 RM/VPDN: Session has been authorized using
dnis:ACME_dnis_numbers
*Mar  1 00:15:53.695: Se0:10 RM/VPDN/session-reply: NAS name HQ-NAS
*Mar  1 00:15:53.699: Se0:10 RM/VPDN/session-reply: Endpoint addresses 172.16.1.9
*Mar  1 00:15:53.703: Se0:10 RM/VPDN/session-reply: VPDN tunnel protocol l2f
*Mar  1 00:15:53.703: Se0:10 RM/VPDN/session-reply: VPDN Group outgoing-2
*Mar  1 00:15:53.707: Se0:10 RM/VPDN/session-reply: VPDN domain dnis:ACME_dnis_numbers
*Mar  1 00:15:53.767: RM/VPDN: MLP Bundle SOHO Session Connect with 1 Endpoints:
*Mar  1 00:15:53.771:   IP 172.16.1.9 OK
*Mar  1 00:15:53.771: RM/VPDN/rm-session-connect/ACME_VPDN: VP
LIMIT/ACTIVE/RESERVED/OVERFLOW are now 6/1/0/0
*Mar  1 00:15:54.815: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:10, changed
state to up
*Mar  1 00:15:57.399: %ISDN-6-CONNECT: Interface Serial0:10 is now connected to SOHO
```

Troubleshooting Customer/VPDN Profile

The following sample output from the **debug resource-pool** command displays when there is no VPDN group associated with an incoming DNIS group. However, the output from the **debug resource-pool** command, as shown here, does not effectively reflect the problem:

```
*Mar  1 03:40:16.483: Se0:15 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 03:40:16.515: Se0:15 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 03:40:16.527: %VPDN-6-AUTHORERR: L2F NAS HQ-NAS cannot locate a AAA server for
Se0:15 user SOHO
*Mar  1 03:40:16.579: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Mar  1 03:40:17.539: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:15, changed
state to up
*Mar  1 03:40:17.615: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
*Mar  1 03:40:19.483: %ISDN-6-CONNECT: Interface Serial0:15 is now connected to SOHO
```

Whenever the **debug resource-pool** command offers no further assistance besides the indication that authorization has failed, enter the **debug aaa authorization** command to further troubleshoot the problem. In this case, the **debug aaa authorization** command output appears as follows:

```
*Mar  1 04:03:49.846: Se0:19 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 04:03:49.854: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): Port='DS0:0:0:0:19'
list='default' service=RM
*Mar  1 04:03:49.858: AAA/AUTHOR/RM vpdn-session: Se0:19 (3912941997) user='301001'
*Mar  1 04:03:49.862: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
service=resource-management
*Mar  1 04:03:49.866: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
protocol=vpdn-session
*Mar  1 04:03:49.866: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-protocol-version=1.0
*Mar  1 04:03:49.870: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-nas-state=3278356
*Mar  1 04:03:49.874: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-call-handle=27
```

```
*Mar  1 04:03:49.878: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
multilink-id=SOHO
*Mar  1 04:03:49.878: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): found list "default"
*Mar  1 04:03:49.882: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): Method=LOCAL
*Mar  1 04:03:49.886: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
service=resource-management
*Mar  1 04:03:49.890: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
protocol=vpdn-session
*Mar  1 04:03:49.890: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-protocol-version=1.0
*Mar  1 04:03:49.894: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-nas-state=3278356
*Mar  1 04:03:49.898: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-call-handle=27
*Mar  1 04:03:49.902: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
multilink-id=SOHO
*Mar  1 04:03:49.906: Se0:19 AAA/AUTHOR/VPDN/RM/LOCAL: Customer ACME has no VPDN  group
for session dnis:ACME_dnis_numbers
*Mar  1 04:03:49.922: Se0:19 AAA/AUTHOR (3912941997): Post authorization status = FAIL
```

Troubleshooting VPDN Profile Limits

The following output from the **debug resource-pool** command displays that VPDN profile limits have been reached:

```
*Mar  1 04:57:53.762: Se0:13 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 04:57:53.774: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 0/0/0/0
*Mar  1 04:57:53.778: RM/VPDN/ACME_VPDN: Session outgoing-2 rejected due to Session Limit
*Mar  1 04:57:53.798: Se0:13 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 04:57:53.802: %VPDN-6-AUTHORFAIL: L2F NAS HQ-NAS, AAA authorization failure for
Se0:13 user SOHO; At Session Max
*Mar  1 04:57:53.866: %ISDN-6-DISCONNECT: Interface Serial0:13  disconnected from  SOHO,
call lasted 2 seconds
*Mar  1 04:57:54.014: %LINK-3-UPDOWN: Interface Serial0:13, changed state to down
*Mar  1 04:57:54.050: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON DS0:0:0:0:13
*Mar  1 04:57:54.054: RM:RPM event call drop
*Mar  1 04:57:54.054: Deallocated resource from res_group isdn-ports
```

Troubleshooting VPDN Group Limits

The following **debug resource-pool** command display shows that VPDN group limits have been reached. From this display, the problem is not obvious. To troubleshoot further, use the **debug aaa authorization** command described in the "Troubleshooting RPMS" section later in this chapter:

```
*Mar  1 05:02:22.314: Se0:17 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 05:02:22.334: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 5/0/0/0
*Mar  1 05:02:22.334: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 05:02:22.358: Se0:17 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 05:02:22.362: %VPDN-6-AUTHORFAIL: L2F NAS HQ-NAS, AAA authorization failure for
Se0:17 user SOHO; At Multilink Bundle Limit
*Mar  1 05:02:22.374: %ISDN-6-DISCONNECT: Interface Serial0:17  disconnected from  SOHO,
call lasted 2 seconds
*Mar  1 05:02:22.534: %LINK-3-UPDOWN: Interface Serial0:17, changed state to down
*Mar  1 05:02:22.570: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON DS0:0:0:0:17
*Mar  1 05:02:22.574: RM:RPM event call drop
*Mar  1 05:02:22.574: Deallocated resource from res_group isdn-ports
```

Troubleshooting VPDN Endpoint Problems

The following output from the **debug resource-pool** command displays that the IP endpoint for the VPDN group is not reachable:

```
*Mar  1 05:12:22.330: Se0:21 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 05:12:22.346: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 5/0/0/0
*Mar  1 05:12:22.350: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 05:12:22.382: Se0:21 RM/VPDN: Session has been authorized using
dnis:ACME_dnis_numbers
*Mar  1 05:12:22.386: Se0:21 RM/VPDN/session-reply: NAS name HQ-NAS
*Mar  1 05:12:22.386: Se0:21 RM/VPDN/session-reply: Endpoint addresses 172.16.1.99
*Mar  1 05:12:22.390: Se0:21 RM/VPDN/session-reply: VPDN tunnel protocol l2f
*Mar  1 05:12:22.390: Se0:21 RM/VPDN/session-reply: VPDN Group outgoing-2
*Mar  1 05:12:22.394: Se0:21 RM/VPDN/session-reply: VPDN domain dnis:ACME_dnis_numbers
*Mar  1 05:12:25.762: %ISDN-6-CONNECT: Interface Serial0:21 is now connected to SOHO
*Mar  1 05:12:27.562: %VPDN-5-UNREACH: L2F HGW 172.16.1.99 is unreachable
*Mar  1 05:12:27.578: RM/VPDN: MLP Bundle SOHO Session Connect with 1 Endpoints:
*Mar  1 05:12:27.582:   IP 172.16.1.99 Destination unreachable
```

Troubleshooting RPMS

In general, the **debug aaa authorization** command is not used for RPM troubleshooting unless the **debug resource-pool** command display is too vague. The **debug aaa authorization** command is more useful for troubleshooting with RPMS. Following is sample output:

```
Router# debug aaa authorization

AAA Authorization debugging is on

Router# show debug

General OS:
  AAA Authorization debugging is on
Resource Pool:
  resource-pool general debugging is on
```

The following output from the **debug resource-pool** and **debug aaa authorization** commands shows a successful RPM connection:

```
*Mar 1 06:10:35.450: AAA/MEMORY: create_user (0x723D24) user='301001'
ruser=''port='DS0:0:0:0:12' rem_addr='102' authen_type=NONE service=NONE priv=0
*Mar 1 06:10:35.462: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907):
Port='DS0:0:0:0:12' list='default' service=RM
*Mar 1 06:10:35.466: AAA/AUTHOR/RM call-accept: DS0:0:0:0:12 (2784758907) user= '301001'
*Mar 1 06:10:35.470: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
service=resource-management
*Mar 1 06:10:35.470: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
protocol=call-accept
*Mar 1 06:10:35.474: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-protocol-version=1.0
*Mar 1 06:10:35.478: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-nas-state=7513368
*Mar 1 06:10:35.482: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-call-type=speech
*Mar 1 06:10:35.486: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-request-type=dial-in
*Mar 1 06:10:35.486: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-link-type=isdn
```

```
*Mar  1 06:10:35.490: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): found list
"default"
*Mar 1 06:10:35.494: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): Method=LOCAL
*Mar 1 06:10:35.498: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received DNIS=301001
*Mar 1 06:10:35.498: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received CLID=102
*Mar 1 06:10:35.502: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received
Port=DS0:0:0:0:12
*Mar 1 06:10:35.506: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
service=resource-management
*Mar 1 06:10:35.510: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
protocol=call-accept
*Mar 1 06:10:35.510: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-protocol-version=1.0
*Mar 1 06:10:35.514: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-nas-state=7513368
*Mar 1 06:10:35.518: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-call-type=speech
*Mar 1 06:10:35.522: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-request-type=dial-in
*Mar 1 06:10:35.526: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-link-type=isdn
*Mar 1 06:10:35.542: AAA/AUTHOR (2784758907): Post authorization status = PASS_REPL
*Mar 1 06:10:35.546: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
service=resource-management
*Mar 1 06:10:35.550: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
protocol=call-accept
*Mar 1 06:10:35.554: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-protocol-version=1.0
*Mar 1 06:10:35.558: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-response-code=overflow
*Mar 1 06:10:35.558: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-handle=47
*Mar 1 06:10:35.562: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-count=2
*Mar 1 06:10:35.566: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-cp-name=ACME
*Mar 1 06:10:35.570: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-rg-name#0=MICA-modems
*Mar 1 06:10:35.574: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-rg-service-name#0=gold
*Mar 1 06:10:35.578: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-treatment=busy
*Mar  1 06:10:35.582: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-type=speech
```

Configuration Examples for RPM

The following sections provide RPM configuration examples:

- Standard Configuration for RPM Example
- Customer Profile Configuration for DoVBS Example
- DNIS Discriminator Profile Example
- CLID Discriminator Profile Example
- Direct Remote Services Configuration Example
- VPDN Configuration Example
- VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example

Standard Configuration for RPM Example

The following example demonstrates a basic RPM configuration:

```
resource-pool enable
resource-pool call treatment resource busy
resource-pool call treatment profile no-answer
!
resource-pool group resource isdn-ports
 range limit 46
resource-pool group resource MICA-modems
 range port 1/0 2/23
!
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
!
resource-pool profile customer DEFAULT
 limit base-size 10
 resource MICA-modems speech service silver
 dnis group default

resource-pool profile discriminator deny_DNIS
 call-type digital
 dnis group bye-bye
!
resource-pool profile service gold
 modem min-speed 33200 max-speed 56000 modulation v90
resource-pool profile service silver
 modem min-speed 19200 max-speed 33200 modulation v34
!
resource-pool aaa protocol local
!
dialer dnis group ACME_dnis_numbers
 number 301001
dialer dnis group bye-bye
 number 301005
```

Tips

- Replace the command string **resource isdn-ports digital** in the previous example with **resource isdn-ports speech** to set up DoVBS. See the section, "Customer Profile Configuration for DoVBS Example," for more information.

  Digital calls to 301001 are associated with the customer ACME by using the resource group "isdn-ports."
- Speech calls to 301001 are associated with the customer ACME by using the resource group "mica-modems" and allow for V.90 connections (anything less than V.90 is also allowed).
- Digital calls to 301005 are denied.
- All other speech calls to any other DNIS number are associated with the customer profile "DEFAULT" by using the resource group "mica-modems" and allow for V.34 connections (anything more than V.34 is not allowed; anything less than V.34 is also allowed).
- All other digital calls to any other DNIS number are not associated with a customer profile and are therefore not allowed.

- The customer profile named “DEFAULT” serves as the default customer profile for speech calls only. If the solution uses an external RPMS server, this same configuration can be used for backup resource pooling if communication is lost between the NAS and the RPMS.

Customer Profile Configuration for DoVBS Example

To allow ISDN calls with a speech bearer capability to be directed to digital resources, make the following change (highlighted in bold) to the configuration shown in the previous section, “Standard Configuration for RPM Example”:

```
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports speech
 dnis group ACME_dnis_numbers
```

This change causes ISDN speech calls (in addition to ISDN digital calls) to be directed to the resource “isdn-ports”; thus, ISDN speech calls provide DoVBS.

DNIS Discriminator Profile Example

The following is sample configuration for a DNIS discriminator. It shows how to enable resource pool management, configure a customer profile, create DNIS groups, and add numbers to the DNIS groups.

```
aaa new-model
!
! Enable resource pool management
resource-pool enable
!
resource-pool group resource digital
 range limit 20
!
! Configure customer profile
resource-pool profile customer cp1
 limit base-size all
 limit overflow-size 0
 resource digital digital
 dnis group ok
!
!
isdn switch-type primary-5ess
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback1
 ip address 192.168.0.0 255.255.255.0
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
```

```
 no peer default ip address
 ppp authentication chap
!
! Configure DNIS groups
dialer dnis group blot
 number 5552003
 number 3456789
 number 2345678
 number 1234567
!
dialer dnis group ok
 number 89898989
 number 5551003
!
dialer-list 1 protocol ip permit
```

CLID Discriminator Profile Example

The following is a sample configuration of a CLID discriminator. It shows how to enable resource pool management, configure resource groups, configure customer profiles, configure CLID groups and DNIS groups, and add them to discriminator profiles.

```
version xx.x
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname cisco-machine
!
aaa new-model
aaa authentication login djm local
!
username eagle password ***
username infiniti 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
!
! Enable resource pool management
resource-pool enable
!
! Configure resource groups
resource-pool group resource digital
 range limit 20
!
! Configure customer profiles
resource-pool profile customer cp1
 limit base-size all
 limit overflow-size 0
 resource digital digital
dnis group ok
!
! Configure discriminator profiles
resource-pool profile discriminator baadaabing
 call-type digital
clid group stompIt
!
```

```
resource-pool profile discriminator baadaaboom
 call-type digital
 clid group splat
!
ip subnet-zero
!
isdn switch-type primary-5ess
chat-script dial ABORT BUSY "" AT OK "ATDT \T" TIMEOUT 30 CONNECT \c
!
!
mta receive maximum-recipients 0
partition flash 2 8 8
!
!
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
!
controller T1 2
 shutdown
 clock source line secondary 2
!
controller T1 3
 shutdown
 clock source line secondary 3
!
controller T1 4
 shutdown
 clock source line secondary 4
!
controller T1 5
 shutdown
 clock source line secondary 5
!
controller T1 6
 shutdown
 clock source line secondary 6
!
controller T1 7
 shutdown
 clock source line secondary 7
!
interface Loopback0
 ip address 192.168.12.1 255.255.255.0
!
interface Loopback1
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback2
 ip address 192.168.17.1 255.255.255.0
!
interface Ethernet0
 ip address 10.0.39.15 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
```

```
interface Serial0
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
 no peer default ip address
 ppp authentication chap pap
!
interface FastEthernet0
 ip address 10.0.38.15 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
 speed 100
!
!
ip local pool default 192.168.13.181 192.168.13.226
ip classless
ip route 172.25.0.0 255.0.0.0 Ethernet0
ip route 172.19.0.0 255.0.0.0 Ethernet0
no ip http server
!
!
! Configure DNIS groups
dialer dnis group blot
 number 4085551003
 number 5552003
 number 2223333
 number 3456789
 number 2345678
 number 1234567
```

```
!
dialer dnis group ok
 number 89898989
 number 4084442002
 number 4085552002
 number 5551003
!
dialer clid group splat
 number 12321224
!
! Configure CLID groups
dialer clid group zot
 number 2121212121
 number 4085552002
!
dialer clid group snip
 number 1212121212
!
dialer clid group stompIt
 number 4089871234
!
dialer clid group squash
 number 5656456
dialer-list 1 protocol ip permit
!
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
line 1 96
 no exec
 exec-timeout 0 0
 autoselect ppp
line aux 0
line vty 0 4
 exec-timeout 0 0
transport input none
!
scheduler interval 1000
end
```

Direct Remote Services Configuration Example

The following example shows a direct remote services configuration:

```
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
 aaa group-configuration tahoe
 source  template acme_direct
!
resource-pool profile customer DEFAULT
 limit base-size 10
 resource MICA-modems speech service silver
 dnis group default
```

```
resource-pool profile discriminator deny_DNIS
  call-type digital
  dnis group bye-bye
!
resource-pool profile service gold
 modem min-speed 33200 max-speed 56000 modulation v90
resource-pool profile service silver
 modem min-speed 19200 max-speed 33200 modulation v34
!
resource-pool aaa protocol local
!
template acme_direct
 peer default ip address pool tahoe
 ppp authentication chap isdn-users
 ppp multilink
!
dialer dnis group ACME_dnis_numbers
 number 301001
dialer dnis group bye-bye
 number 301005
```

VPDN Configuration Example

Adding the following commands to those listed in the section "Standard Configuration for RPM Example" earlier in this chapter allows you to use VPDN by setting up a VPDN profile and a VPDN group:

Note If the limits imposed by the VPDN profile are not required, do not configure the VPDN profile. Replace the **vpdn profile** *ACME_VPDN* command under the customer profile ACME with the **vpdn group** *outgoing-2* command.

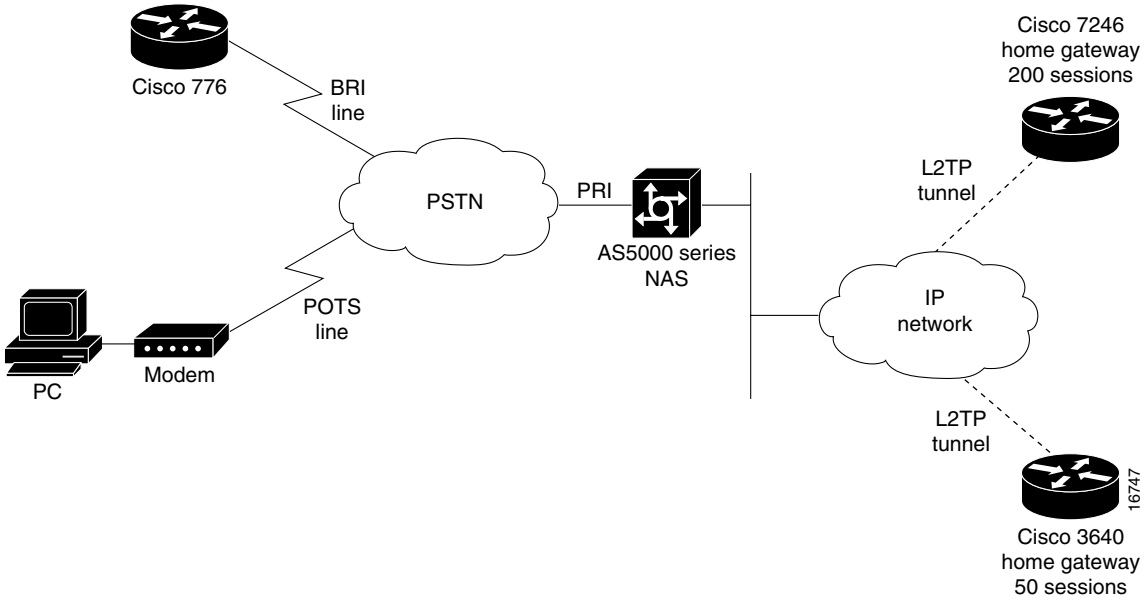
```
resource-pool profile vpdn ACME_VPDN
 limit base-size 6
 limit overflow-size 0
 vpdn group outgoing-2
!
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
!
vpdn profile ACME_VPDN
!
vpdn enable
!
vpdn-group outgoing-2
 request dialin
  protocol 12f
  dnis ACME_dnis_numbers
 local name HQ-NAS
 initiate-to ip 172.16.1.9
 multilink bundle 1
 multilink link 2
!
dialer dnis group ACME_dnis_numbers
 number 301001
```

VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example

Cisco IOS software enables you to balance and back up VPDN sessions across multiple tunnel endpoints (HGW/LNS). When a user or session comes into the NAS/LAC, a VPDN load-balancing algorithm is triggered and applied to the call. The call is then passed to an available HGW/LNS. You can modify this function by limiting the number of sessions supported on an HGW/LNS router and limiting the number of MLP bundles and links.

Figure 109 shows an example of one NAS/LAC that directs calls to two HGW/LNS routers by using the L2TP tunneling protocol. Each router has a different number of supported sessions and works at a different speed. The NAS/LAC is counting the number of active simultaneous sessions sent to each HGW/LNS.

Figure 109 Home Gateway Load Sharing and Backup



In a standalone NAS environment (no RPMS server used), the NAS has complete knowledge of the status of tunnel endpoints. Balancing across endpoints is done by a “least-filled tunnel” or a “next-available round robin” approach. In an RPMS-controlled environment, RPMS has the complete knowledge of tunnel endpoints. However, the NAS still has the control over those tunnel endpoints selected by RPMS.

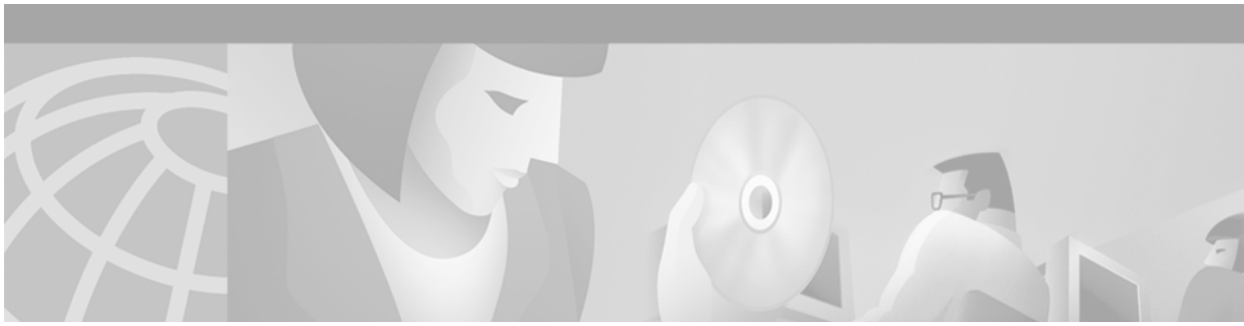
A standalone NAS uses the following default search criteria for load-balancing traffic across multiple endpoints (HGW/LNS):

- Select any idle endpoint—an HGW/LNS with no active sessions.
- Select an active endpoint that currently has a tunnel established with the NAS.
- If all specified load-sharing routers are busy, select the backup HGW. If all endpoints are busy, report that the NAS cannot find an IP address to establish the call.

Note This default search order criteria is independent of the Cisco RPMS application scenario. A standalone NAS uses a different load-sharing algorithm than the Cisco RPMS. This search criteria will change as future enhancements become available.

The following is an example of VPDN load sharing between multiple HGW/LNSs:

```
vpdn enable
!
vpdn-group outgoing-2
 request dialin
  protocol l2tp
  dnis ACME_dnis_numbers
 local name HQ-NAS
 initiate-to ip 172.16.1.9
 loadsharing ip 172.16.1.9 limit 200
 loadsharing ip 172.16.2.17 limit 50
 backup ip 172.16.3.22
```



Configuring Wholesale Dial Performance Optimization

This chapter describes the Wholesale Dial Performance Optimization feature in the following sections:

Note This task provides inbound and outbound performance optimization for wholesale dial customers who provide ports to America Online (AOL). It is configured only on Cisco AS5800 access servers.

Wholesale Dial Performance Optimization Feature Overview

Both the inbound and outbound aspects of this feature are enabled using the **autocommand-options telnet-faststream** command.

- Outbound—Provides stream processing, allowing the output data processing to occur at the interrupt level. Being event driven, this removes polling and process switching overhead. In addition, the flow control algorithm is enhanced to handle the higher volume of traffic and to eliminate some out-of-resource conditions that could result in abnormal termination of the session.
- Inbound—Provides stream processing with the same improvements as for outbound traffic. Also, it removes scanning for special escape characters in the data stream; this is very process-intensive and is not required for this application. (In other situations, the escape characters allow for a return to the privileged EXEC mode prompt (#) on the router.) In addition, Nagle's algorithm is used to form the inbound data stream into larger packets, thus minimizing packet-processing overhead.

This configuration is designed to provide more efficiency in the data transfers for AOL port suppliers who are using a Cisco network access server to communicate with a wholesale dial carrier.

The Cisco AS5800 access server is required to support all dial-in lines supported by two complete T3 connections (that is, 1344 connections) running TCP Clear connections to an internal host. The desired average data throughput for these connections is 6 kbps outbound and 3 kbps inbound.

When using the **autocommand-options telnet-faststream** command, no special character processing, including break recognition, is performed on incoming data from the dial shelf. This requires the TCP Clear connection to run as the sole connection on the TTY line. This sole connection is terminated by TTY line termination or TCP connection termination, with no EXEC session capability for the user. This

has been implemented by specifying a new **autocommand-options telnet-faststream** command that, in conjunction with the **autocommand telnet** command with the **/stream** option, enables Telnet faststream processing. This capability is also available for TACACS/RADIUS attribute-value pair processing, because this processing uses the **autocommand** facility.

How to Configure Automatic Command Execution

The following are three options for configuring the **autocommand telnet /stream** line configuration command:

- Automatic command execution can be configured on the lines.
- Automatic command execution can be configured using user ID and password.
- Automatic command execution can also be configured at a TACACS/RADIUS server, if the username authentication is to be performed there, rather than on the router.

To configure automatic command execution on the lines of a Cisco AS5800 universal network access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line 1/3/00 1/11/143`	Selects the lines to be configured and begins line configuration mode.
Step 2	`Router(config-line)# autocommand telnet aol-host 5190 /stream`	Configures autocommand on the lines.

To configure automatic command execution using a user ID and password on a Cisco AS5800 universal network access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# username aol password aol`	Defines the user ID and password.
Step 2	`Router(config)# username aol autocommand telnet aol-host 5190 /stream`	Configures autocommand on the user ID.

You can also configure automatic command execution at a TACACS/RADIUS server if the username authentication is to be performed there rather than on the router. The AV-pair processing allows autocommand to be configured.

How to Configure TCP Clear Performance Optimization

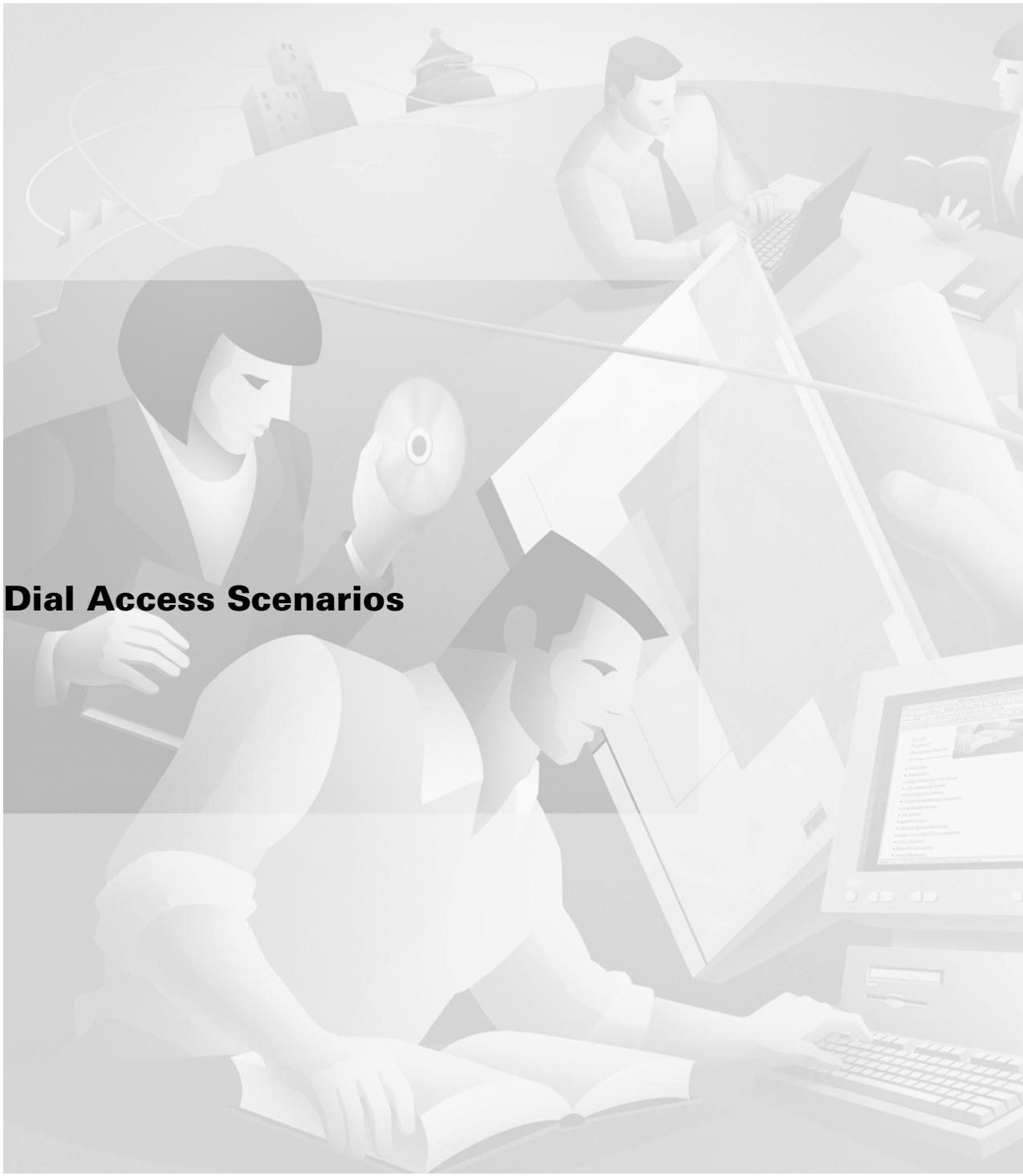
To enable TCP Clear performance optimization, automatic command execution must be configured to enable Telnet faststream capability. To implement TCP Clear performance optimization on a Cisco AS5800 universal network access server, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# line 1/3/00 1/11/143`	Selects the lines to be configured and begins line configuration mode.
Step 2	`Router(config-line)# autocommand telnet-faststream`	Enables the TCP Clear performance optimization on the selected lines.

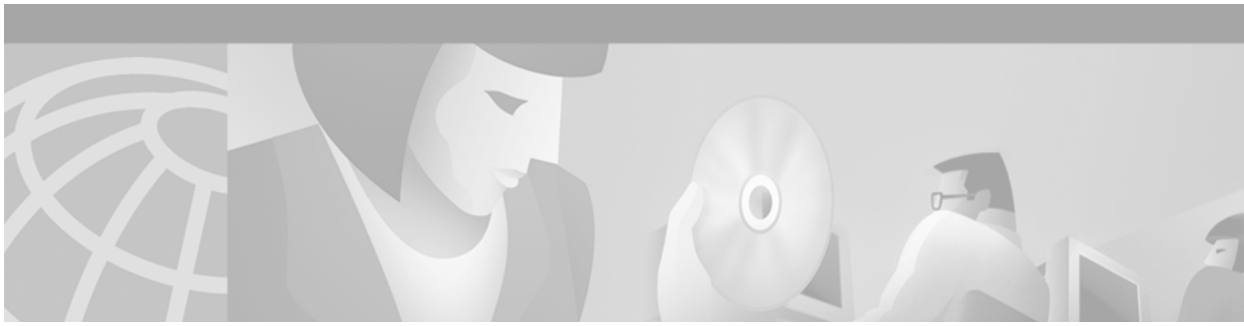
Verifying Configuration of TCP Clear Performance Optimization

To check for correct configuration, use the **show line** command. In the following example, Telnet faststream is enabled under "Capabilities".

```
Router# show line 1/4/00

   Tty Typ     Tx/Rx    A Modem  Roty AccO AccI   Uses   Noise  Overruns   Int
*   1/4/00 Digital modem - inout     -    -    -      1       0     0/0       -

Line 1/4/00, Location: "", Type: ""
Length: 24 lines, Width: 80 columns
Status: PSI Enabled, Ready, Connected, Active, No Exit Banner
  Modem Detected
Capabilities: Hardware Flowcontrol In, Hardware Flowcontrol Out
  Modem Callout, Modem RI is CD, Line usable as async interface
  Hangup on Last Close, Modem Autoconfigure, Telnet Faststream
Modem state: Ready
Modem hardware state: CTS DSR  DTR RTS
modem=1/4/00, vdev_state(0x00000000)=CSM_OC_STATE, bchan_num=(T1 1/2/0:7:20)
vdev_status(0x00000001): VDEV_STATUS_ACTIVE_CALL.

Group codes:    0, Modem Configured
Special Chars: Escape  Hold  Stop  Start  Disconnect  Activation
                ^^x    none   -     -       none
Timeouts:      Idle EXEC    Idle Session   Modem Answer  Session   Dispatch
                never         never                      none     not set
                            Idle Session Disconnect Warning
                              never
                            Login-sequence User Response
                             00:00:30
                            Autoselect Initial Wait
                              not set
Modem type is 9600.
Session limit is not set.
Time since activation: never
Editing is enabled.
History is enabled, history size is 10.
DNS resolution in show commands is enabled
Full user help is disabled
Allowed transports are telnet.  Preferred is lat.
Automatically execute command "telnet 10.100.254.254 2145 /stream"
No output characters are padded
```



Dial Access Scenarios



Dial Networking Business Applications

This chapter provides an introduction to common dial networking scenarios used by service providers and enterprises and includes the following sections:

- Dial Networking for Service Providers and Enterprises
- Common Dial Applications
- IP Address Strategies

Providing dial access means to set up one or more access servers or routers to allow on-demand connectivity for individual remote nodes or remote offices. The dial network solutions described in this chapter are based on business case scenarios. Depending on your business application, dial access has different implementations.

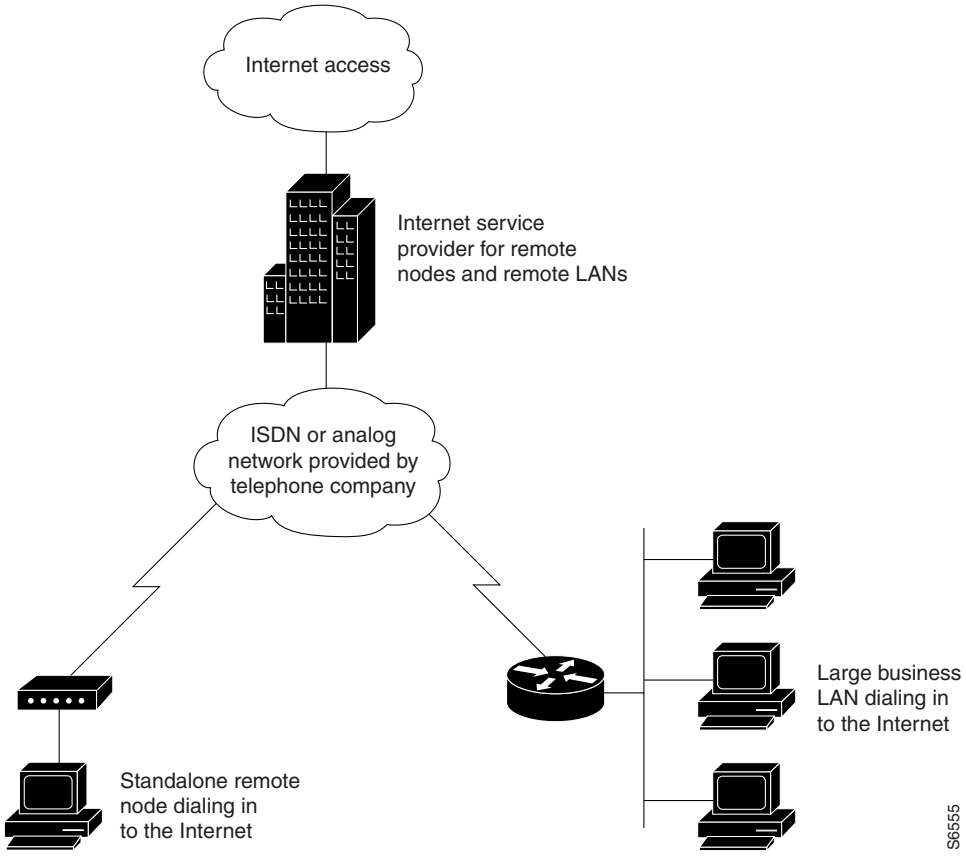
Dial Networking for Service Providers and Enterprises

Service providers tend to supply public and private dial-in services for businesses or individual home users. Enterprises tend to provide private dial-in access for employees dialing in from remote LANs (such as a remote office) or individual remote nodes (such as a telecommuter). Additionally, there are hybrid forms of dial access—virtual private dialup networks (VPDNs)—that are jointly owned, operated, and set up by both service providers and enterprises.

Figure 110 displays a common dial topology used by an Internet service provider (ISP). The central dial-in site is owned and controlled by the ISP, who only accepts dial-in calls. Enterprises and individual remote clients have no administrative control over the point of presence (POP) of the ISP.

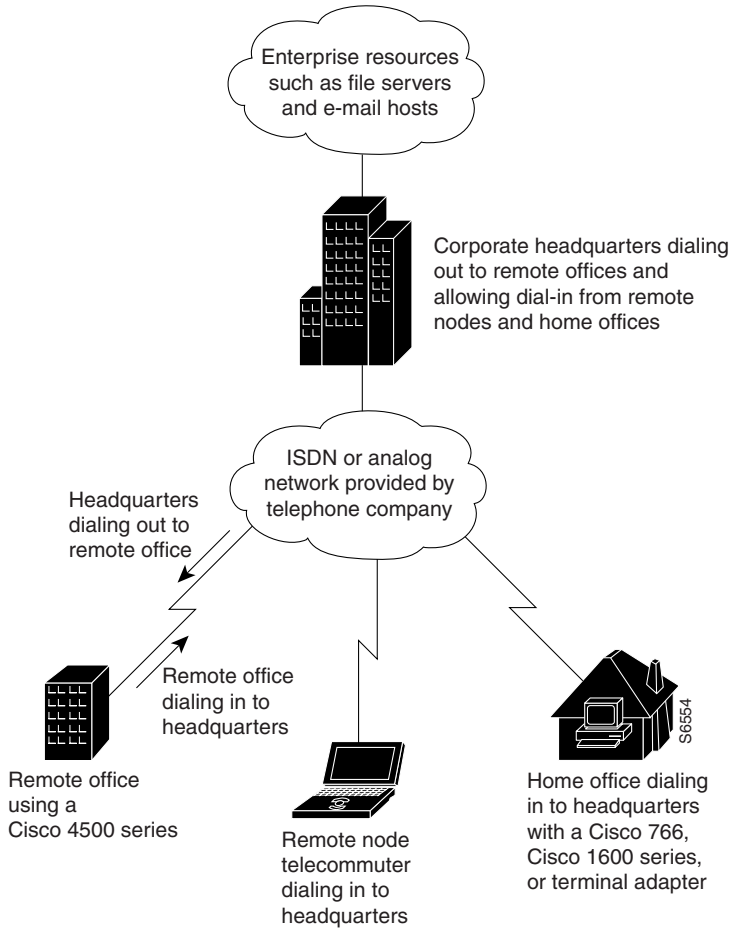
Note Many additional dial network strategies exist for different business applications. This overview is intended to provide only a sample of the most common dial business needs as experienced by the Cisco dial escalation team.

Figure 110 Sample Dial Network for an ISP



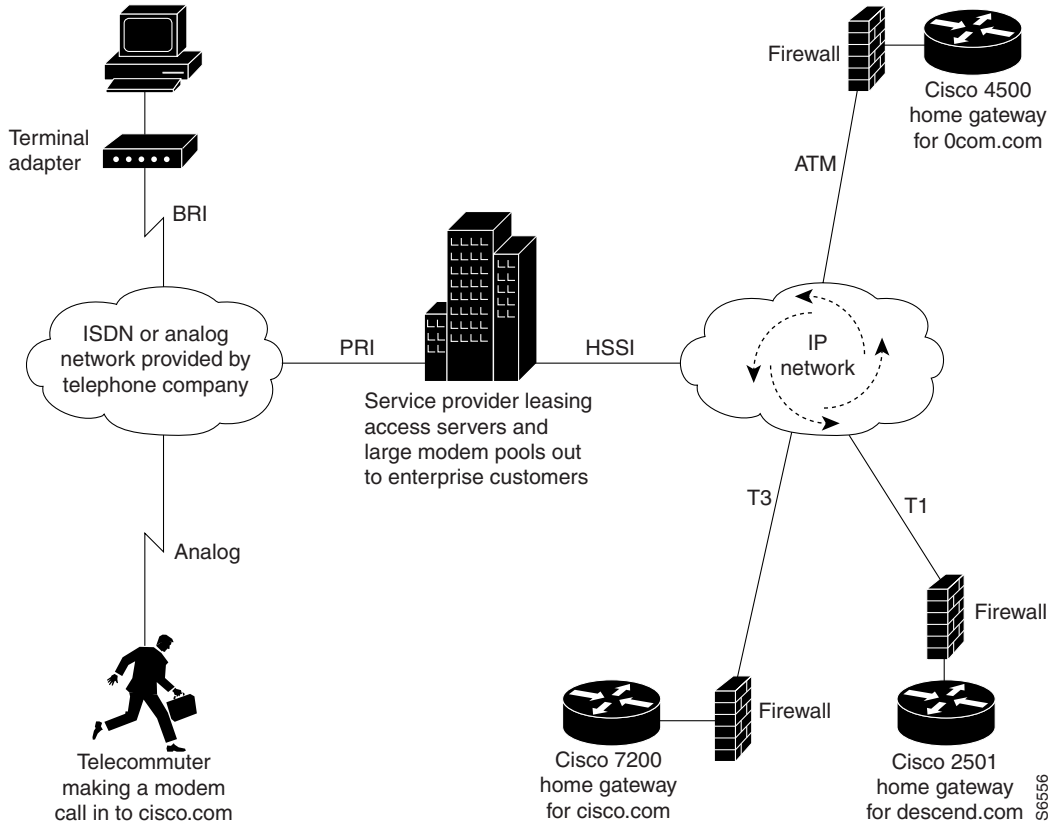
Enterprises can provide bidirectional access services with remote LANs and one-way dial-in access for standalone remote nodes. Bidirectional access means that remote LANs can dial in to the enterprise, and the enterprise can dial out to the remote LANs. A remote LAN can be a large remote office or a small home office. A standalone remote node can be an individual PC that is dynamically assigned an IP address from the modem pool of the enterprise. In most cases, an enterprise has complete administrative control over its local and remote devices. (See Figure 111.)

Figure 111 Sample Dial Network for an Enterprise



Service providers and enterprises both benefit from a hybrid dial solution called VPDN. Service providers offer virtually private access to enterprises by providing the dial-in access devices for the enterprise to use (for example, access servers and modem pools). In this solution, service providers construct the networking fabric for city-to-city dial connectivity for the enterprise. Enterprises provide only a home gateway router (with no attached modems) and a WAN connection to their service provider. VPDN dial solutions enable the enterprise to continue to maintain complete administrative control over its remote locations and network resource privileges. (See Figure 112.)

Figure 112 Sample VPDN for Service Providers and Enterprises



Common Dial Applications

The hardware and software configuration designs for dial networks are derived from business operations needs. This section describes several of the most common business dial scenarios that Cisco Systems is supporting for basic IP and security services.

Refer to the scenario that best describes your business or networking needs:

- The following dial scenarios are commonly used by service providers. For detailed description and configuration information, see the chapter "Telco and ISP Dial Scenarios and Configurations" later in this manual.
  - Scenario 1, Small- to Medium-Scale POPs
    (one or two access servers at the central dial-in site)
  - Scenario 2, Large-Scale POPs
    (more than two access servers at the central dial-in site, Multichassis Multilink PPP or MMP)
  - Scenario 3, PPP Calls over X.25 Networks
- The following dial scenarios are commonly used by enterprises. For detailed description and configuration information, see the chapter "Enterprise Dial Scenarios and Configurations."
  - Scenario 1, Remote Offices and Telecommuters Dialing In to a Central Site
  - Scenario 2, Bidirectional Dial Between Central Sites and Remote Offices
  - Scenario 3, Telecommuters Dialing In to a Mixed Protocol Environment

IP Address Strategies

Exponential growth in the remote access router market has created new addressing challenges for ISPs and enterprise users. Companies that use dial technologies seek addressing solutions that will:

- Minimize Internet access costs for remote offices
- Minimize configuration requirements on remote access routers
- Enable transparent and dynamic IP address allocation for hosts in remote environments
- Improve network security capabilities at each remote small office, home office site
- Conserve registered IP addresses
- Maximize IP address manageability

Remote networks have variable numbers of end systems that need access to the Internet; therefore, some ISPs are interested in allocating just one IP address to each remote LAN.

In enterprise networks where telecommuter populations are increasing in number, network administrators need solutions that ease configuration and management of remote routers and provide conservation and dynamic allocation of IP addresses within their networks. These solutions are especially important when network administrators implement large dial-up user pools where ISDN plays a major role.

Choosing an Addressing Scheme

Use an IP addressing scheme that is appropriate for your business scenario as described in the following sections:

- Classic IP Addressing
- Cisco Easy IP

Additionally, here are some addressing issues to keep in mind while you evaluate different IP address strategies:

- How many IP addresses do you need?
- Do you want remote clients to dial in to your network and connect to server-based services, which require statically assigned IP addresses?
- Is your primary goal to provide Internet services to a network (for example, surfing the web, downloading e-mail, using TCP/IP applications)?
- Can you conduct business with only a few registered IP addresses?
- Do you need a single contiguous address space, or can you function with two non-contiguous address spaces?

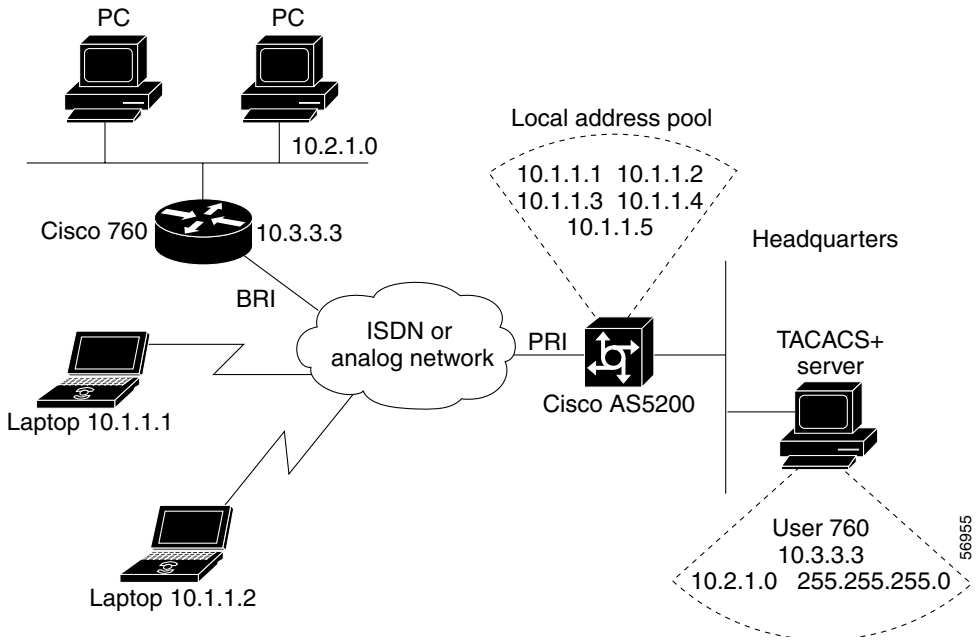
Classic IP Addressing

This section describes two classic IP addressing strategies that you can use to set up dial-in access. Classic IP addresses are statically or dynamically assigned from your network to each site router or dial-in client. The IP address strategy you use depends on whether you are allowing remote LANs or individual remote clients to dial in.

A remote LAN usually consists of a single router at the gateway followed by multiple nodes such as 50 PCs. The IP address on the gateway router is fixed or statically assigned (for example, 3.3.3.3). This device always uses the address 3.3.3.3 to dial in to the enterprise or service provider network. There is also a segment or subnet associated with the gateway router (for example, 2.1.1.0 255.255.255.0), which is defined by the dial-in security server.

For individual remote clients dialing in, a specific range or pool of IP addresses is defined by the gateway access server and dynamically assigned to each node. When a remote node dials in, it receives an address from the specified address pool. This pool of addresses usually resides locally on the network access server. Whereas, the remote LANs have predefined or statically assigned addresses. The accompanying subnet is usually statically assigned too. (See Figure 113.)

Figure 113 Classic IP Address Allocation



Here are some advantages and disadvantages of manually assigning IP addresses:

- Advantages
  - Web servers or Xservers can be stationed at remote locations.
  - Since addresses are members of your network, they are perfectly transparent.
- Disadvantages
  - IP address assignments can be difficult to administer or manage. You may also need to use complicated subnetting configurations.
  - Statically assigned IP addresses use up precious address space.
  - Strong routing configuration skills are usually required.

Cisco Easy IP

Two of the key problems facing the Internet are depletion of IP address space and scaling in routing. The Cisco Easy IP feature combines Network Address Translation (NAT) and PPP/Internet Protocol Control Protocol (IPCP). This feature enables a Cisco router to automatically negotiate its own registered WAN

interface IP address from a central server and allows all remote hosts to access the global Internet using this single registered IP address. Because Cisco Easy IP uses existing port-level multiplexed NAT functionality within the Cisco IOS software, IP addresses on the remote LAN are invisible to the Internet.

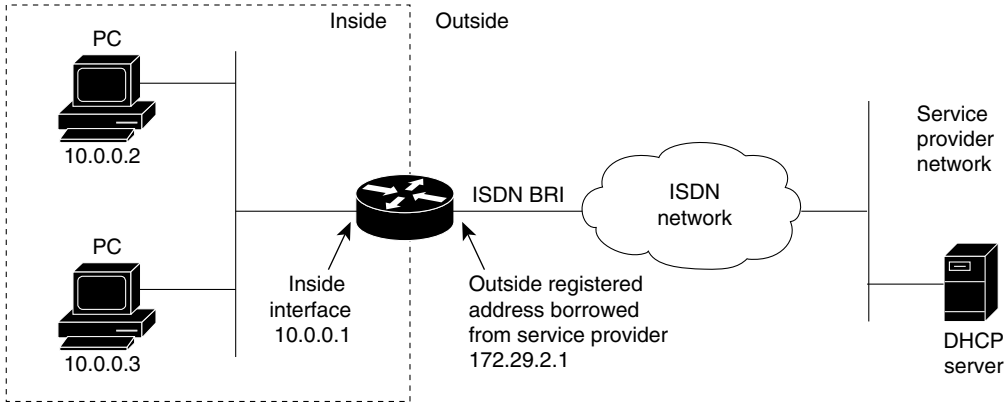
Cisco Easy IP Component Technologies

Cisco Easy IP solution is a scalable, standards-based, "plug-and-play" solution that comprises a combination of the following technologies:

- NAT—Described in RFC 1631. NAT operates on a router that usually connects two or more networks together. Using Cisco Easy IP, at least one of these networks (designated as "inside" or "LAN") is addressed with private (RFC 1918) addresses that must be converted into a registered address before packets are forwarded onto the other registered network (designated as "outside" or "WAN"). Cisco IOS software provides the ability to define one-to-one translations (NAT) as well as many-to-one translations (Port Address Translation [PAT]). Within the context of Cisco Easy IP, PAT is used to translate all internal private addresses to a single outside registered IP address.
- PPP/IPCP—Defined in RFC 1332. This protocol enables users to dynamically configure IP addresses over PPP. A Cisco Easy IP router uses PPP/IPCP to dynamically negotiate its own WAN interface address from a central access server or DHCP server.

Figure 114 shows an example of how Cisco Easy IP works. A range of registered or unregistered IP addresses are used inside a company's network. When a dial-up connection is initiated by an internal node, the router uses the Cisco Easy IP feature to rewrite the IP header belonging to each packet and translate the private address into the dynamically assigned and registered IP address, which could be borrowed from a service provider.

Figure 114 Translating and Borrowing IP Addresses



NAT table

Inside IP Address	Outside Address
10.0.0.2	172.29.2.1: 4011
10.0.0.3	172.29.2.1: 4012

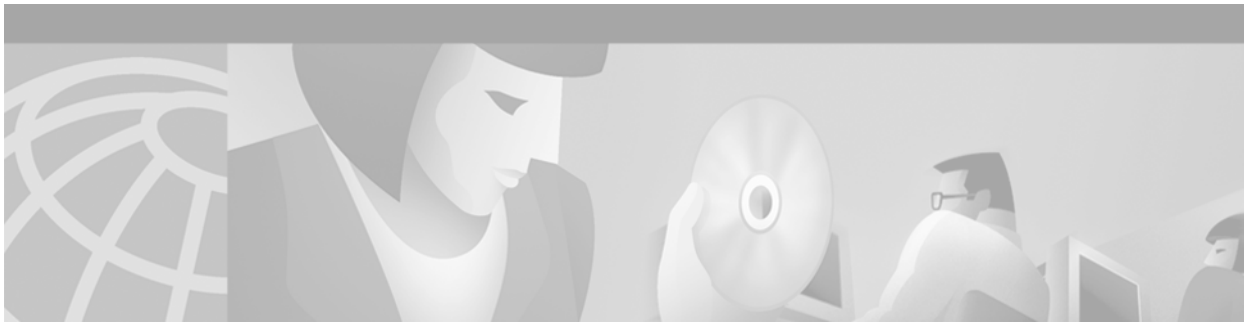
54717

For a more detailed description of how Cisco Easy IP works, see the chapter "Configuring Cisco Easy IP."

Key Benefits of Using Cisco Easy IP

The Cisco Easy IP feature provides the following benefits:

- Reduces Internet access costs by using dynamically allocated IP addresses. Using dynamic IP address negotiation (PPP/IPCP) at each remote site substantially reduces Internet access costs. Static IP addresses cost more to *purchase* compared to dynamically allocated or *rented* IP addresses. Cisco Easy IP enables you to rent IP addresses. In addition, dynamically assigned IP addresses saves you time and money associated with subnet mask configuration tasks on hosts. It also eliminates the need to configure host IP addresses when moving from network to network.
- Simplifies IP address management. Cisco Easy IP enables ISPs to allocate a single registered IP address to each remote LAN. Because only a single registered IP address is required to provide global Internet access to all users on an entire remote LAN, customers and ISPs can use their registered IP addresses more efficiently.
- Conserves registered IP addresses. Suppose you want to connect to the Internet, but not all your hosts have globally unique IP addresses. NAT enables private IP internetworks that use nonregistered or overlapping IP addresses to connect to the Internet. NAT is configured on the router at the border of a stub domain (referred to as the *inside network*) and a public network such as the Internet (referred to as the *outside network*). The private addresses you set up on the inside of your network translate in to a *single* registered IP addresses on the outside of your network.
- Provides remote LAN IP address privacy. Because Cisco Easy IP uses existing port-level multiplexed NAT functionality within Cisco IOS software, IP addresses on the remote LAN are invisible to the Internet, making the LAN inherently more secure. As seen by the external network, the source IP address of all traffic from the remote LAN is the single registered IP address of the WAN interface for the Cisco Easy IP router.



Enterprise Dial Scenarios and Configurations

This chapter provides sample configurations for specific dial scenarios used by enterprise networks (not telephone companies or Internet service providers). Each configuration is designed to support IP network traffic with basic security for the specified scenario.

The following scenarios are described:

- Scenario 1—Remote Offices and Telecommuters Dialing In to a Central Site
- Scenario 2—Bidirectional Dial Between Central Sites and Remote Offices
- Scenario 3—Telecommuters Dialing In to a Mixed Protocol Environment

Note If you use Token card-based security in your dial network, we recommend that you enable Password Authentication Protocol (PAP) authentication and disable the Multilink protocol to maximize dial-in performance.

Remote User Demographics

Employees stationed in remote offices or disparate locations often dial in to central sites or headquarter offices to download or upload files and check e-mail. These employees often dial in to the corporate network from a remote office LAN using ISDN or from another location such as a hotel room using a modem.

The following remote enterprise users typically dial in to enterprise networks:

- Full-time telecommuters—Employees using stationary workstations to dial in from a small office, home office (SOHO), making ISDN connections with terminal adapters or PC cards through the public telephone network, and operating at higher speeds over the network, which rules out the need for a modem.
- Travelers—Employees such as salespeople that are not in a steady location for more than 30 percent of the time usually dial in to the network with a laptop and modem through the public telephone network, and primarily access the network to check E-mail or transfer a few files.
- Workday extenders—Employees that primarily work in the company office, occasionally dial in to the enterprise with a mobile or stationary workstation plus modem, and primarily access the network to check E-mail or transfer a few files.

Demand and Scalability

You need to evaluate scalability and design issues before you build a dial enterprise network. As the number of company employees increases, the number of remote users who need to dial in increases. A good dial solution scales upward as the demand for dial-in ports grows. For example, it is not uncommon for a fast-growing enterprise to grow from a demand of 100 modems to 250 modems in less than one year.

You should always maintain a surplus of dial-in ports to accommodate company growth and occasional increases in access demand. In the early stages of a fast-growing company that has 100 modems installed for 6000 registered remote users, only 50 to 60 modems might be active at the same time. As demand grows over one year, 250 modems might be installed to support 10,000 registered token card holders.

During special company occasions, such as worldwide conventions, demand for remote access can also increase significantly. During such activities, dial-in lines are used heavily throughout the day and evening by remote sales people using laptops to access E-mail and share files. This behavior is indicative of sales people working away from their home territories or sales offices. Network administrators need to prepare for these remote access bursts, which cause significant increases for remote access demand.

Remote Offices and Telecommuters Dialing In to a Central Site

Remote office LANs typically dial in to other networks using ISDN. Remote offices that use Frame Relay require a more costly dedicated link.

Connections initiated by remote offices and telecommuters are brought up on an as-needed basis, which results in substantial cost savings for the company. In dial-on-demand scenarios, users are not connected for long periods of time. The number of remote nodes requiring access is relatively low, and the completion time for the dial-in task is short.

Central sites typically do not dial out to the remote LANs. Instead, central sites respond to calls. Remote sites initiate calls. For example, a field sales office might use ISDN to dial in to and browse a central site's intranet. Additionally a warehouse comprising five employees can use ISDN to log in to a remote network server to download or upload product order information. For an example of bidirectional dialing, see the section "Bidirectional Dial Between Central Sites and Remote Offices" later in this chapter.

Note Dial-on-demand routing (DDR) uses static routes or snapshot routing. For IP-only configurations, static routes are commonly used for remote dial-in. For Internet Protocol Exchange (IPX) networking, snapshot routing is often used to minimize configuration complexity.

Network Topologies

Figure 115 shows an example of a remote office that places digital calls in to a central site network. The remote office router can be any Cisco router with a BRI physical interface, such as a Cisco 766 or Cisco 1604 router. The central office gateway router can be any Cisco router that supports PRI connections, such as a Cisco 3600 series, Cisco 4000 series, or Cisco 7000 series router.

Figure 115 Remote Office Dialing In to a Central Site

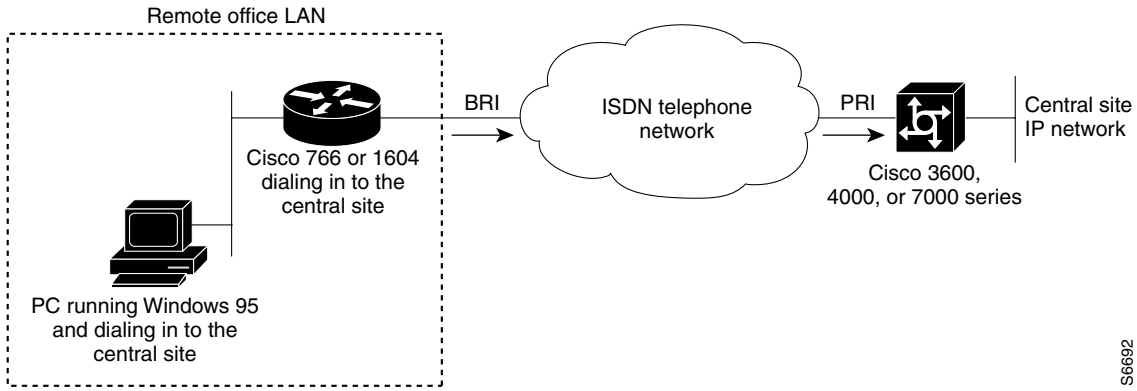
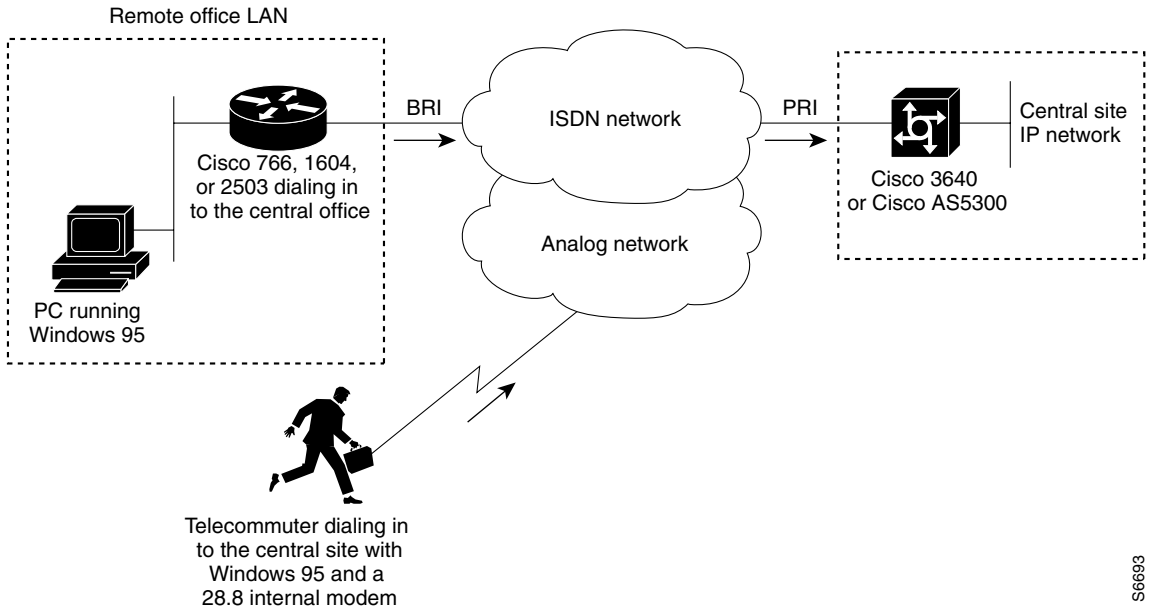


Figure 116 shows an example of a remote office and telecommuter dialing in to a central site. The remote office places digital calls. The telecommuter places analog calls. The remote office router can be any Cisco router with a BRI interface, such as a Cisco 766, Cisco 1604, or Cisco 2503 router. The central office gateway router is a Cisco AS5300 series access server or a Cisco 3640 router, which supports both PRI and analog connections.

Figure 116 Remote Office and Telecommuter Dialing In to a Central Site



Dial-In Scenarios

The configuration examples in the following sections provide different combinations of dial-in scenarios, which can be derived from Figure 115 and Figure 116:

- Cisco 1604 Remote Office Router Dialing In to a Cisco 3620 Access Router
- Remote Office Router Dialing In to a Cisco 3620 Router

- Cisco 700 Series Router Using Port Address Translation to Dial In to a Cisco AS5300 Access Server
- Cisco 3640 Central Site Router Configuration to Support ISDN and Modem Calls
- Cisco AS5300 Central Site Configuration Using Remote Security

Note Be sure to include your own IP addresses, host names, and security passwords where appropriate if you use these examples in your own network.

Cisco 1604 Remote Office Router Dialing In to a Cisco 3620 Access Router

This section provides a common configuration for a Cisco 1604 remote office router dialing in to a Cisco 3620 access router positioned at a central enterprise site. Only ISDN digital calls are supported in this scenario. No analog modem calls are supported. All calls are initiated by the remote router on an as-needed basis. The Cisco 3620 router is not set up to dial out to the Cisco 1604 router. (Refer to Figure 115.)

The Cisco 1604 and Cisco 3620 routers use the IP unnumbered address configurations, MLP, and the dial-load threshold feature, which brings up the second B channel when the first B channel exceeds a certain limit. Because static routes are used, a routing protocol is not configured. A default static route is configured on the Cisco 1604 router, which points back to the central site. The central site also has a static route that points back to the remote LAN. Static route configurations assume that you have only one LAN segment at each remote office.

Cisco 1604 Router Configuration

The following configuration runs on the Cisco 1604 router, shown in Figure 115. This SOHO router places digital calls in to the Cisco 3620 central site access router. See the next example for the running configuration of the Cisco 3620 router.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname remotelan1
!
enable secret cisco
!
username NAS password dialpass
username admin password cisco

isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 10.2.1.1 255.255.255.0
!
interface BRI0
 ip unnumbered Ethernet0
 encapsulation ppp
 dialer map ip 10.1.1.10 name NAS 5551234
 dialer load-threshold 100 either
 dialer-group 1
 no fair-queue
 ppp authentication chap pap callin
 ppp multilink
```

```
!
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.10
ip route 10.1.1.10 255.255.255.255 BRI0
dialer-list 1 protocol ip permit
!
line con 0
line vty 0 4
 login local
!
end
```

Cisco 3620 Router Configuration

The following sample configuration runs on the Cisco 3620 router shown in Figure 115. This modular access router has one 2-port PRI network module installed in slot 1 and one 1-port Ethernet network module installed in slot 0. The router receives only digital ISDN calls from the Cisco 1604 router. The configuration for the Cisco 1604 router was provided in the previous example.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
enable secret cisco
!
username admin password cisco
username remotelan1 password dialpass

async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
controller T1 1/0
 framing esf
 clock source line
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1/1
 framing esf
 clock source line
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet 0/0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
```

```
interface Serial 1/0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial 1/1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 default-metric 64 100 250 100 1500
 redistribute static
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1

ip route 10.2.1.1 255.255.255.255 Dialer0
ip route 10.2.1.0 255.255.255.0 10.2.1.1

ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Remote Office Router Dialing In to a Cisco 3620 Router

This section provides a common configuration for a Cisco 700 or 800 series remote office router placing digital calls in to a Cisco 3620 router positioned at a central enterprise site. All calls are initiated by the remote router on an as-needed basis. The Cisco 3620 router is not set up to dial out to the remote office router. (See Figure 115.)

Cisco 700 Series Router Configuration

The following configuration task is for a Cisco 700 series ISDN router placing digital calls in to a central site router that supports ISDN PRI, such as the Cisco 3620 router. In this scenario, ISDN unnumbered interfaces with static routes are pointing back to the Cisco 3620.

To configure the router, use the following commands in EXEC mode. However, this configuration assumes that you are starting from the router's default configuration. To return the router to its default configuration, issue the **set default** command.

	Command	Purpose

Step 1	`>`
	`> set systemname remotelan1`
	`remotelan1>`
		At the system prompt level, specifies the host name of the router, which is also used when responding to Challenge Handshake Authentication Protocol (CHAP) authentication with the Cisco 3620. For CHAP authentication, the system's name must match the username configured on the Cisco 3620.

Step 2	`remotelan1> set ppp secret client`
	`remotelan1> Enter new password: dialpass`
	`remotelan1> Enter new password: dialpass`
		Sets the transmit and receive password for the client. This is the password which is used in response to CHAP authentication requests, and it must match the username password configured on the Cisco 3620 router.

Step 3	`remotelan1> set encapsulation ppp`	Sets PPP encapsulation for incoming and outgoing authentication instead of CPP.
Step 4	`remotelan1> set ppp multilink on`	Enables Multilink PPP (MLP).
Step 5	`remotelan1> set user nas` `remotelan1> New user nas being created`	Creates the profile named nas, which is reserved for the Cisco 3620 router.
Step 6	`remotelan1:nas> set ip 0.0.0.0`	Specifies the LAN IP address. The sequence 0.0.0.0 means that it will use the address assigned to it from the central Cisco 3620 router. See Step 14.
Step 7	`remotelan1:nas> set ip framing none`	Configures the profiles to not use Ethernet framing.
Step 8	`remotelan1:nas> set ip route destination 0.0.0.0 gateway 10.1.1.10`	Sets the default route to point to the Ethernet IP address of the Cisco 3620 router.
Step 9	`remotelan1:nas> set timeout 300`	Sets the idle time at which the B channel will be dropped. In this case, the line is dropped after 300 seconds of idle time.
Step 10	`remotelan1:nas> set 1/2 number 5551234`	Sets the number to call when dialing out of the first and second B channel.
Step 11	`remotelan1:nas> cd lan`	Enters LAN profile mode.
Step 12	`remotelan1:LAN> set bridging off`	Turns bridging off.
Step 13	`remotelan1:LAN> set ip routing on`	Turns on IP routing.
Step 14	`remotelan1:LAN> set ip address 10.2.1.1`	Sets the LAN IP address for the interface.

After you configure the Cisco 760 or Cisco 770 series router, the final configuration should resemble the following:

```
set systemname remotelan1
set ppp secret client
set encapsulation ppp
set ppp multilink on
cd lan
set bridging off
set ip routing on
set ip 10.2.1.1
set subnet 255.255.255.0
set user nas
set bridging off
set ip 0.0.0.0
set ip netmask 0.0.0.0
set ip framing none
set ip route destination 0.0.0.0 gateway 10.1.1.10
set timeout 300
set 1 number 5551234
set 2 number 5551234
```

The previous software configuration does not provide for any access security. To provide access security, use the following optional commands in EXEC mode:

Command	Purpose
Router> **set ppp authentication incoming chap**	Provides CHAP authentication to incoming calls.
Router> **set callerid**	Requires the calling parties number to be matched against the configured receive numbers (such as set by the **set callidreceive #** command). This command also denies all incoming calls if no callidreceive number is configured.
Router> **set remoteaccess protected**	Specifies a remote system password, which enables you to make changes on the router from a remote location.
Router> **set localaccess protected**	Specifies a local system password, which enables you to make changes on the router from a local console connection.
Router> **set password system**	Sets the system password for the previous access configurations.

Cisco 3620 Router Configuration

The following example provides a sample configuration for the Cisco 3620 router. This modular access router has one 2-port PRI network module installed in slot 1 and one 1-port Ethernet network module installed in slot 0. The router receives only digital ISDN calls over T1 lines from the Cisco 700 series remote office router, which was described in the previous example.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
hostname NAS
!
aaa new-model
aaa authentication login default local
```

```
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
enable secret cisco
!
username admin password cisco
username remotelan1 password dialpass
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
controller T1 1/0
 framing esf
 clock source line
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1/1
 framing esf
 clock source line
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet 0/0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial 1/0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial 1/1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
```

```
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 default-metric 64 100 250 100 1500
 redistribute static
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1

ip route 10.2.1.1 255.255.255.255 Dialer0
ip route 10.2.1.0 255.255.255.0 10.2.1.1

ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Cisco 700 Series Router Using Port Address Translation to Dial In to a Cisco AS5300 Access Server

This section shows a Cisco 700 series router using the port address translation (PAT) feature to dial in to a Cisco AS5300 central site access server. IP addresses are assigned from the central site, which leverages the PAT feature to streamline multiple devices at the remote site through a single assigned address. In this example, the Cisco 700 series router has a private range of IP addresses used on the Ethernet side. However, the router is able to translate between the local private addresses and the dynamically registered address on the WAN interface. (See Figure 115.)

Cisco 700 Series Configuration

The sample configuration in this section allows PCs on a LAN to boot up and acquire their IP address dynamically from a Cisco 700 series router, which in turn translates the private addresses into a single IP address assigned from a Cisco AS5300 central site router. The Cisco 700 series router also passes information via DHCP regarding the Domain Name System (DNS) server (in this example, 10.2.10.1) and the Windows Internet naming service (WINS) server (in this example, 10.2.11.1) along with the domain name.

A possible sequence of events would be a remote PC running Windows 95 boots up on the Ethernet segment and gets its IP address and network information from the Cisco 700 series router. The PC then opens up Netscape and attempts to view a web page at the central site, which causes the router to dial in to the central site. The router dynamically obtains its address from the central site pool of addresses and uses it to translate between the private address on the local Ethernet segment and the registered IP address borrowed from the central site router.

To configure a remote router, use the following commands beginning in EXEC mode:

	Command	Purpose
Step 1	> > **set systemname remotelan1** Router>	At the system prompt level, specifies the host name of the router, which is also used when responding to CHAP authentication with the Cisco 3620 router. For CHAP authentication, the system's name must match the username configured on the Cisco 3620.
Step 2	Router> **set ppp secret client** Router> Enter new password: **dialpass** Router> Enter new password: **dialpass**	Sets the transmit and receive password for the client. This is the password which is used in response to CHAP authentication requests, and it must match the username password configured on the Cisco 3620 router.
Step 3	Router> **set encapsulation ppp**	Sets PPP encapsulation for incoming and outgoing authentication instead of CPP.
Step 4	Router> **set ppp multilink on**	Enables MLP.
Step 5	Router> **set dhcp server**	Enables the router to act as a DHCP server and assign addresses from the private network. By default, all DHCP client addresses are assigned from the 10.0.0.0 network.
Step 6	Router> **set dhcp dns primary 10.2.10.1**	Passes the DNS server IP address to the DHCP client.
Step 7	Router> **set dhcp wins 10.2.11.1**	Passes the IP address of the WINS server to the DHCP client.
Step 8	Router> **set dhcp domain nas.com**	Sets the DHCP domain name for the Cisco 3620 central site router.
Step 9	Router> **set user nas** Router> New user nas being created	Creates the profile named nas, which is setup for the Cisco 3620 router.
Step 10	Router:nas> **set ip pat on**	Enables Port Address Translation (PAT) on the router.
Step 11	Router:nas> **set ip framing none**	Configures the profiles to not use Ethernet framing.
Step 12	Router:nas> **set ip route destination 0.0.0.0 gateway 10.1.1.0**	Sets the default route to point to the Ethernet IP address of Cisco 3620 router.
Step 13	Router:nas> **set 1 number 5551234**	Sets the number to call when dialing out of the first B channel.
Step 14	Router:nas> **set 2 number 5551234**	Sets the number to call when dialing out of the second B channel.
Step 15	Router:nas> **cd lan**	Enters LAN profile mode.
Step 16	Router:LAN> **set bridging off**	Turns bridging off.
Step 17	Router:LAN> **set ip routing on**	Turns IP routing on.

After you configure the router, the configuration should resemble the following:

```
set systemname remotelan1
set encapsulation ppp
set ppp secret client
set ppp multilink on
set dhcp server
set dhcp dns primary 10.2.10.1
set dhcp wins 10.2.11.1
set dhcp domain nas.com
set user nas
set bridging off
```

```
set ip routing on
set ip framing none
set ip pat on
set ip route destination 0.0.0.0 gateway 10.1.1.0
set 1 number 5551234
set 2 number 5551234
```

Cisco AS5300 Router Configuration

The following example configures a Cisco AS5300 router for receiving calls from the router in the previous example.

Note This configuration can also run on a Cisco 4000, Cisco 3600, or Cisco 7000 series router. However, the interface numbering scheme for these routers will be in the form of slot/port. Additionally, the clocking will be set differently. Refer to your product configuration guides and configuration notes for more details.

```
!
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
enable secret cisco
!
username admin password cisco
username remotelan1 password dialpass
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
```

```
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 default-metric 64 100 250 100 1500
 redistribute static
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1

ip route 10.2.1.1 255.255.255.255 Dialer0
ip route 10.2.1.0 255.255.255.0 10.2.1.1

ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line aux 0
 login authentication console
```

```
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

In this configuration, the local pool is using a range of unused addresses on the same subnet on which the Ethernet interface is configured. The addresses will be used for the remote devices dialing in to the Cisco AS5300 access server.

Cisco 3640 Central Site Router Configuration to Support ISDN and Modem Calls

The following configuration allows remote LANs and standalone remote users with modems to dial in to a central site. Figure 116 shows the network topology.

The Cisco 3640 router has the following hardware configuration for this scenario:

- One 2-port ISDN-PRI network module installed in slot 1.
- One digital modem network module installed in slot 2 and slot 3.
- One 1-port Ethernet network module installed in slot 0.

Note Each MICA technologies digital modem card has its own group async configuration. Additionally, a single range of asynchronous lines is used for each modem card. For additional interface numbering information, refer to the document *Digital Modem Network Module Configuration Note*.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
enable secret cisco
!
username admin password cisco
username remotelan1 password dialpass1
username remotelan2 password dialpass2
username PCuser1 password dialpass3
username PCuser2 password dialpass4
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
controller T1 1/0
 framing esf
 clock source line
 linecode b8zs
 pri-group timeslots 1-24
!
```

```
controller T1 1/1
 framing esf
 clock source line
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0/0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial 1/0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial 1/1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 65 88
!
interface Group-Async2
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 97 120
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
```

```
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line 65 88
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line 97 120
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Cisco AS5300 Central Site Configuration Using Remote Security

The previous examples in this section configured static CHAP authentication on the central router using the **username** command. A more common configuration to support modem and ISDN calls on a single chassis is to use the AAA security model and an external security server at the central site. We recommend that you have a solid understanding of basic security principles and the AAA model before you set up this configuration. For more information about security, see the *Cisco IOS Security Configuration Guide*.

Central Site Cisco AS5300 Configuration Using TACACS+ Authentication

The following example assumes that you are running TACACS+ on the remote security server:

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login console enable
aaa authentication login vty tacacs+
aaa authentication login dialin tacacs+
aaa authentication ppp default tacacs+
aaa authentication ppp dialin if-needed tacacs+
enable secret cisco
!
```

```
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
```

```
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 redistribute static
 default-metric 64 100 250 100 1500
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
end
```

TACACS+ Security Server Entry

The following example can be configured on a remote TACACS+ security server, which complements the Cisco AS5300 access server configuration listed in the previous example:

```
user = remotelan1 {
           chap = cleartext "dialpass1"
           service = ppp protocol = ip {
                       addr = 10.2.1.1
                       route = "10.2.1.0 255.255.255.0"
           }
}


user = PCuser1 {
           login = cleartext "dialpass2"
           chap = cleartext "dialpass2"
           service = ppp protocol = ip {
                       addr-pool = dialin_pool
           }
           service = exec {
                       autocmd = "ppp negotiate"
           }
}

user = PCuser2 {
           login = cleartext "dialpass3"
           chap = cleartext "dialpass3"
           service = ppp protocol = ip {
                       addr-pool = dialin_pool
           }
```

```
service = exec {
            autocmd = "ppp negotiate"
}
```

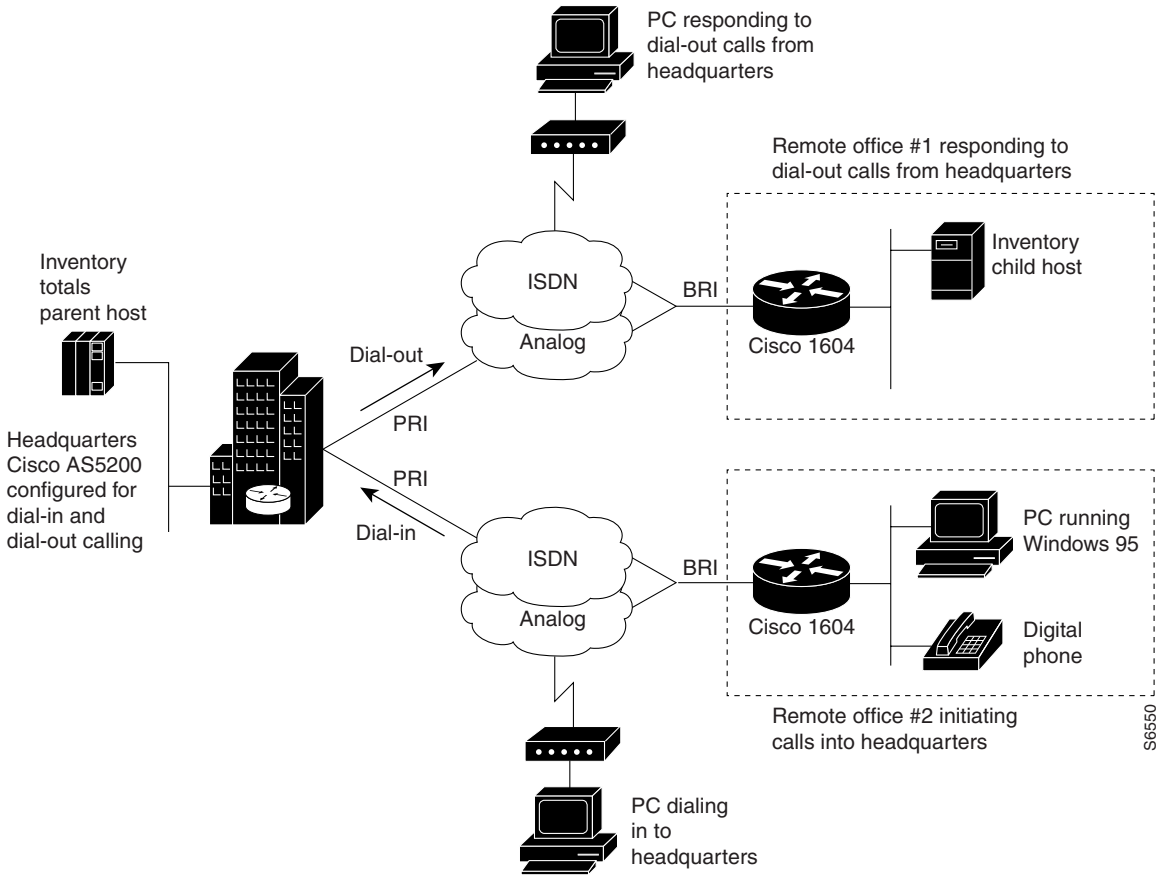
Bidirectional Dial Between Central Sites and Remote Offices

Sometimes a gateway access server at headquarters is required to dial out to a remote site while simultaneously receiving incoming calls. This type of network is designed around a specific business support model.

Dial-In and Dial-Out Network Topology

Figure 117 shows a typical dial-in and dial-out network scenario, which amounts to only 25 percent of all dial topologies. The Cisco AS5300 access server at headquarters initiates a connection with a Cisco 1604 router at remote office 1. After a connection is established, the file server at the remote site (shown as Inventory child host) runs a batch processing application with the mainframe at headquarters (shown as Inventory totals parent host). While files are being transferred between remote office 1 and headquarters, remote office 2 is successfully dialing in to headquarters.

Figure 117 Headquarters Configured for Dial-In and Dial-out Networking



There are some restrictions for dial-out calling. Dial-out analog and digital calls are commonly made to remote ISDN routers, such as the Cisco 1604 router. On the whole, dial out calls are not made from a central site router to a remote PC but rather from a remote PC in to the central site. However, central site post offices often call remote office routers on demand to deliver E-mail. Callback is enabled on dial-in scenarios only. The majority of a dial out software configuration is setup on the router at headquarters, not the remote office router. Dialing out to a stack group of multiple chassis is not supported by Cisco IOS software. Note that Multichassis Multilink PPP (MMP) and virtual private dialup networks (VPDNs) are dial-in only solutions.

Dialer Profiles and Virtual Profiles

Profiles are set up to discriminate access on a user-specific basis. For example, if the chief network administrator is dialing in to the enterprise, a unique user profile can be created with an idle timeout of one year, and universal access privileges to all networks in the company. For less fortunate users, access can be restricted to an idle timeout of 10 seconds and network connections setup for only a few addresses.

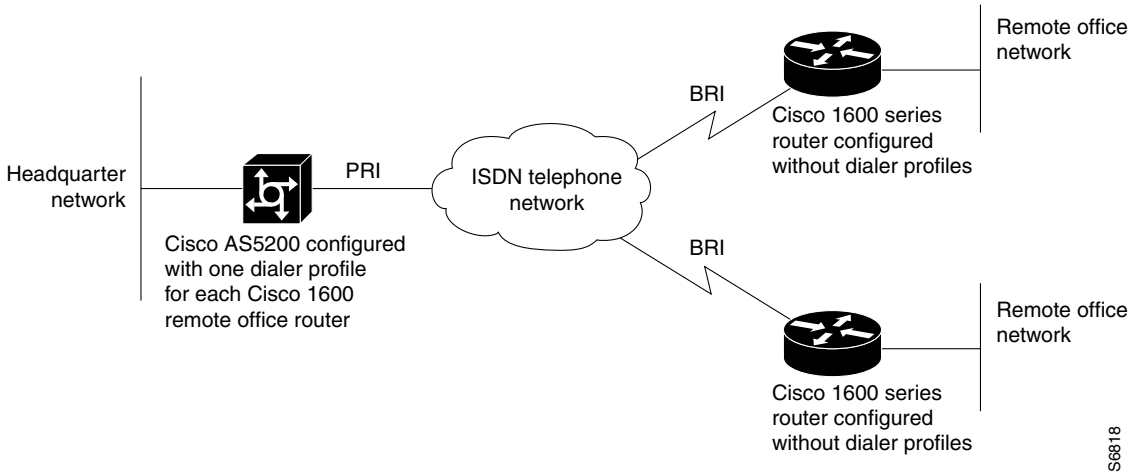
Depending on the size and scope of your dial solution, you can set up two different types of profiles: dialer profiles or virtual profiles. Dialer profiles are individual user profiles set up on routers or access servers in a small-scale dial solution. This type of profile is configured locally on the router and is limited by the number of interfaces that exist on the router. When an incoming call comes into the dial pool, the dialer interface binds the caller to a dialer profile via the caller ID or the caller name.

Figure 118 shows an example of how dialer profiles can be used when:

- You need to bridge over multiple ISDN channels.
- You want to use ISDN to back up a WAN link, but still have the ISDN interface available during those times that the WAN link is up.
- A security server, such as a AAA TACACS or RADIUS server, is not available for use.

Note For more information about dialer profiles, see the chapters "Configuring Peer-to-Peer DDR with Dialer Profiles" and "Configuring Dial Backup with Dialer Profiles."

Figure 118 Dial-In Scenario for Dialer Profiles



Virtual profiles are user-specific profiles for large-scale dial solutions; however, these profiles are not manually configured on each router or access server. A virtual profile is a unique PPP application that can create and configure a virtual access interface dynamically when a dial-in call is received, and tear down the interface dynamically when the call ends.

The configuration information for a virtual access interface in a virtual profile can come from the virtual template interface, or from user-specific configuration information stored on an AAA server, or both. The virtual profile user-specific configuration stored on the AAA server is identified by the authentication name for the call-in user. (That is, if the AAA server authenticates the user as samson, the user-specific configuration is listed under samson in the AAA user file.) The virtual profile user-specific configuration should include only the configuration that is not shared by multiple users. Shared configuration should be placed in the virtual template interface, where it can be cloned on many virtual access interfaces as needed.

AAA configurations are much easier to manage for large numbers of dial-in users. Virtual profiles can span across a group of access servers, but a AAA server is required. Virtual profiles are set up independently of which access server, interface, or port number users connect to. For users that share duplicate configuration information, it is best to enclose the configuration in a virtual template. This requirement eliminates the duplication of commands in each of the user records on the AAA server.

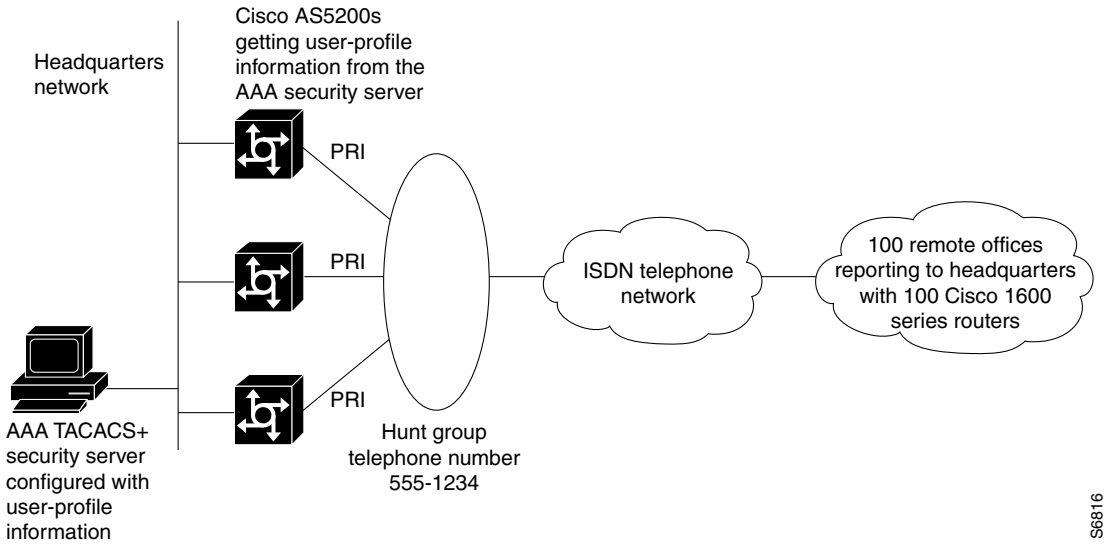
The user-specific AAA configuration used by virtual profiles is interface configuration information and downloaded during link control protocol (LCP) negotiations. Another feature, called per-user configuration, also uses configuration information gained from a AAA server. However, per-user configuration uses *network* configuration (such as access lists and route filters) downloaded during NCP negotiations.

Figure 119 shows an example of how virtual profiles are used:

- A large-scale dial-in solution is available, which includes many access servers or routers (for example, three or more devices stacked together in an MMP scenario).
- Discrimination between large numbers of users is needed.
- Setup and maintenance of a user profile for each dial-in user on each access server or router is much too time consuming.
- A security server, such as a AAA TACACS or RADIUS server, is available for use.

Note For a virtual profile configuration example, see the section "Large-Scale Dial-In Configuration Using Virtual Profiles" later in this chapter. For more information about virtual profiles, see the chapters "Configuring Virtual Profiles" and "Configuring Per-User Configuration" in this publication.

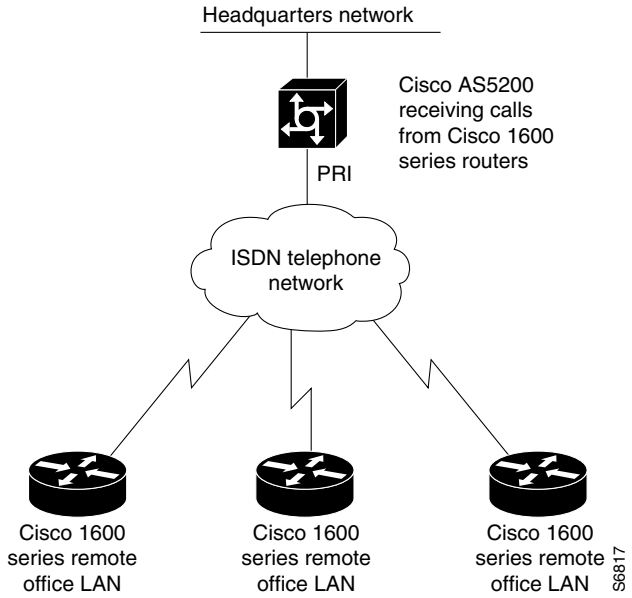
Figure 119 Dial-In Scenario for Virtual Profiles



Running Access Server Configurations

In most cases, dialer profiles are configured on access servers or routers that receive calls and must discriminate between users, such as many different remote routers dialing in. (See Figure 120.)

Figure 120 Remote Cisco 1600s Dialing In to a Cisco AS5300 at the Central Site



Access servers or routers that only place calls (not receive calls) do not need any awareness of configured dialer profiles. Remote routers do not need to discriminate on the basis of which device they are calling in to. For example, if multiple Cisco 1600 series routers are dialing in to one Cisco AS5300 access

server, the Cisco 1600 series routers should not be configured with dialer profiles. The Cisco AS5300 access server should be configured with dialer profiles. Do not configure dialer profiles on devices that *only* make calls.

The configurations examples in the following section are provided for different types of dial scenarios, which can be derived from Figure 117 through Figure 120:

- Examples with dialer profiles:
  - Cisco AS5300 Access Server Configuration with Dialer Profiles
  - Cisco 1604 ISDN Router Configuration with Dialer Profiles
  - Cisco 1604 Router Asynchronous Configuration with Dialer Profiles
- Examples without dialer profiles:
  - Cisco AS5300 Access Server Configuration Without Dialer Profiles
  - Cisco 1604 ISDN Router Configuration Without Dialer Profiles
  - Cisco 1604 Router Asynchronous Configuration Without Dialer Profiles
- Large-Scale Dial-In Configuration Using Virtual Profiles

Note Be sure to include your own IP addresses, host names, and security passwords where appropriate if configuring these examples in your network.

Cisco AS5300 Access Server Configuration with Dialer Profiles

The following bidirectional dial configuration runs on the Cisco AS5300 access server at headquarters in Figure 117. This configuration enables calls to be sent to the SOHO router and received from remote hosts and clients. The calling is bidirectional.

```
version xx.x
service udp-small-servers
service tcp-small-servers
!
hostname 5300
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
enable secret cisco
!
username async1 password cisco
username async2 password cisco
username async3 password cisco
username async4 password cisco
username async5 password cisco
username async6 password cisco
username async7 password cisco
username async8 password cisco
username isdn1 password cisco
username isdn2 password cisco
username isdn3 password cisco
username isdn4 password cisco
username isdn5 password cisco
```

```
username isdn6 password cisco
username isdn7 password cisco
username isdn8 password cisco
username DialupAdmin password cisco
!
isdn switch-type primary-dms100
chat-script cisco-default ABORT ERROR "" "AT" OK "ATDT\T" TIMEOUT 60 CONNECT
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface loopback 1
 ip address 172.18.38.40 255.255.255.128
!
interface loopback 2
 ip address 172.18.38.130 255.255.255.128
!
interface Ethernet0
 ip address 172.18.39.40 255.255.255.0
 no ip mroute-cache
 ip ospf priority 0
!
interface Serial0:23
 no ip address
 no ip mroute-cache
 encapsulation ppp
 isdn incoming-voice modem
 dialer pool-member 2
!
interface Serial1:23
 no ip address
 no ip mroute-cache
 encapsulation ppp
 isdn incoming-voice modem
 dialer pool-member 2
!
interface Group-Async1
 no ip address
 no ip mroute-cache
 encapsulation ppp
 async mode interactive
 dialer in-band
 dialer pool-member 1
 ppp authentication chap pap
 group-range 1 48
!
interface Dialer10
 ip unnumbered loopback 1
 encapsulation ppp
 peer default ip address dialin_pool
 dialer remote-name async1
 dialer string 14085268983
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
```

```
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer11
 ip unnumbered loopback 1
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async2
 dialer string 14085262012
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer12
 ip unnumbered loopback 1
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async3
 dialer string 14085260706
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer13
 ip unnumbered loopback 1
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async4
 dialer string 14085262731
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer14
 ip unnumbered loopback 1
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async5
 dialer string 14085264431
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer15
 ip unnumbered loopback 1
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async6
 dialer string 14085261933
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
```

```
interface Dialer16
 ip unnumbered loopback 1
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async7
 dialer string 14085267631
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer17
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name async8
 dialer string 14085265153
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer18
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn1
 dialer string 14085267887
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer19
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn2
 dialer string 14085261591
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer20
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn3
 dialer string 14085262118
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer21
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn4
 dialer string 14085263757
 dialer hold-queue 10
 dialer pool 2
```

```
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer22
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn5
 dialer string 14085263769
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer23
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn6
 dialer string 14085267884
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer24
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn7
 dialer string 14085267360
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
interface Dialer25
 ip unnumbered loopback 2
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name isdn8
 dialer string 14085260361
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
router ospf 1
redistribute static subnets
passive-interface Dialer1
passive-interface Dialer2
network 172.18.0.0 0.0.255.255 area 0
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip domain-name cisco.com
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line 1 24
 no exec
 exec-timeout 0 0
```

```
 autoselect during-login
 autoselect ppp
 script dialer cisco-default
 login local
 modem InOut
 modem autoconfigure type microcom_hdms
 transport input telnet
line aux 0
line vty 0 1
 exec-timeout 60 0
 password cisco
 login
line vty 2 5
 exec-timeout 5 0
 password cisco
 login
!
end
```

Cisco 1604 ISDN Router Configuration with Dialer Profiles

The following configuration runs on the remote office Cisco 1604 router, which receives calls from the Cisco AS5300 central site access server. (See Figure 117.)

```
version xx.x
service udp-small-servers
service tcp-small-servers
!
hostname isdn1
!
enable password cisco
!
username 5300 password cisco
username isdn1 password cisco
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.18.40.1 255.255.255.0
!
interface BRI0
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap pap
!
interface Dialer1
 ip address 172.18.38.131 255.255.255.128
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name 5300
 dialer string 14085269328
 dialer hold-queue 10
 dialer pool 2
 dialer-group 1
 ppp authentication chap pap
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.38.130
dialer-list 1 protocol ip permit
!
line con 0
line vty 0 4
```

```
 password cisco
 login
 password cisco
 login
!
end
```

Cisco 1604 Router Asynchronous Configuration with Dialer Profiles

The following asynchronous configuration runs on the remote office Cisco 1604 router, which receives calls from the Cisco AS5300 central site access server. (See Figure 117.)

```
version xx.x
service udp-small-servers
service tcp-small-servers
!
hostname async1
!
enable password cisco
!
username 5300 password cisco
username async1 password cisco
chat script dial_out "" "ATDT\T" timeout 60 connect \c
!
interface Ethernet0
 ip address 172.18.41.1 255.255.255.0
!
interface serial 0
 physical-layer async
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap pap
!
interface Dialer10
 ip address 172.18.38.41 255.255.255.128
 encapsulation ppp
 no peer default ip address pool
 dialer remote-name 5300
 dialer string 14085269328
 dialer hold-queue 10
 dialer pool 1
 dialer-group 1
 ppp authentication chap pap
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.38.40
dialer-list 1 protocol ip permit
!
line con 0
line 1
password cisco
login
script modem dial_out
!
end
```

Cisco AS5300 Access Server Configuration Without Dialer Profiles

The following bidirectional dial configuration runs on the Cisco AS5300 access server at headquarters in Figure 117. This configuration enables calls to be sent to the SOHO router and received from remote hosts and clients. The calling is bidirectional.

```
version xx.x
service udp-small-servers
service tcp-small-servers
!
hostname 5300
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
enable secret cisco
!
username async1 password cisco
username async2 password cisco
username async3 password cisco
username async4 password cisco
username async5 password cisco
username async6 password cisco
username async7 password cisco
username async8 password cisco
username isdn1 password cisco
username isdn2 password cisco
username isdn3 password cisco
username isdn4 password cisco
username isdn5 password cisco
username isdn6 password cisco
username isdn7 password cisco
username isdn8 password cisco
username DialupAdmin password cisco
!
isdn switch-type primary-dms100
chat-script cisco-default ABORT ERROR "" "AT" OK "ATDT\T" TIMEOUT 60 CONNECT
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
 description ISDN Controller 0
!
controller T1 1
 framing esf
 clock source line secondary
 linecode b8zs
 pri-group timeslots 1-24
 description ISDN Controller 1
!
interface Ethernet0
 ip address 172.18.39.40 255.255.255.0
 no ip mroute-cache
 ip ospf priority 0
!
```

```
interface Serial0:23
 no ip address
 no ip mroute-cache
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 2
!
interface Serial1:23
 no ip address
 no ip mroute-cache
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 2
!
interface Group-Async1
 no ip address
 no ip mroute-cache
 encapsulation ppp
 async dynamic address
 async mode interactive
 dialer in-band
 dialer rotary-group 1
 ppp authentication pap callin
 ppp pap sent-username HQ5300 password 7 09434678520A
 group-range 1 24
!
interface Dialer1
 ip address 172.18.38.40 255.255.255.128
 encapsulation ppp
 no peer default ip address pool
 dialer in-band
 dialer map ip 172.18.38.41 name async1 14445558983
 dialer map ip 172.18.38.42 name async2 14445552012
 dialer map ip 172.18.38.43 name async3 14445550706
 dialer map ip 172.18.38.44 name async4 14445552731
 dialer map ip 172.18.38.45 name async5 14445554431
 dialer map ip 172.18.38.46 name async6 14445551933
 dialer map ip 172.18.38.47 name async7 14445557631
 dialer map ip 172.18.38.48 name async8 14445555153
 dialer hold-queue 10
 dialer-group 1
 ppp authentication pap chap callin
 ppp pap sent-username DialupAdmin password 7 07063D11542
!
interface Dialer2
 ip address 172.18.38.130 255.255.255.128
 encapsulation ppp
 no peer default ip address pool
 dialer in-band
 dialer map ip 172.18.38.131 name isdn1 14445557887
 dialer map ip 172.18.38.132 name isdn2 14445551591
 dialer map ip 172.18.38.133 name isdn3 14445552118
 dialer map ip 172.18.38.134 name isdn4 14445553757
 dialer map ip 172.18.38.135 name isdn5 14445553769
 dialer map ip 172.18.38.136 name isdn6 14445557884
 dialer map ip 172.18.38.137 name isdn7 14445557360
 dialer map ip 172.18.38.138 name isdn8 14445550361
 dialer hold-queue 10
 dialer-group 1
 ppp authentication chap pap
 ppp multilink
!
```

```
router ospf 1
 redistribute static subnets
 passive-interface Dialer1
 passive-interface Dialer2
 network 172.18.0.0 0.0.255.255 area 0
!
ip domain-name cisco.com
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line 1 24
 no exec
 exec-timeout 0 0
 autoselect during-login
 autoselect ppp
 script dialer cisco-default
 login local
 modem InOut
 modem autoconfigure type microcom_hdms
 transport input telnet
line aux 0
line vty 0 1
 exec-timeout 60 0
 password cisco
 login
line vty 2 5
 exec-timeout 5 0
 password cisco
 login
!
end
```

Cisco 1604 ISDN Router Configuration Without Dialer Profiles

The following configuration runs on the remote office Cisco 1604 router, which dials in to the Cisco AS5300 access server at headquarters in Figure 117. This configuration does not receive calls from the Cisco AS5300 access server.

```
!
version 11.1
service udp-small-servers
service tcp-small-servers
!
hostname isdn1
!
enable password cisco
!
username 5300 password cisco
username isdn1 password cisco
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.18.40.1 255.255.255.0
!
interface BRI0
 ip address 172.18.38.131 255.255.255.128
 encapsulation ppp
 dialer map ip 172.18.38.130 name 5300 14085269328
```

```
 dialer-group 1
 ppp authentication chap pap
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.38.130
dialer-list 1 protocol ip permit
!
line con 0
line vty 0 4
 password cisco
 login
 password cisco
 login
!
end
```

Cisco 1604 Router Asynchronous Configuration Without Dialer Profiles

The following asynchronous configuration runs on the remote office Cisco 1604 router, which dials in to the Cisco AS5300 access server at headquarters in Figure 117. This configuration does not receive calls from the Cisco AS5300 access server.

```
version xx.x
service udp-small-servers
service tcp-small-servers
!
hostname async1
!
enable password cisco
!
username 5300 password cisco
username async1 password cisco
chat script dial_out "" "ATDT\T" timeout 60 connect \c
!
interface Ethernet0
 ip address 172.18.41.1 255.255.255.0
!
interface serial 0
 physical-layer async
 ip address 172.18.38.41 255.255.255.128
 encapsulation ppp
 dialer in-band
 dialer map ip 172.18.38.40 name 5300 modem-script dial_out 14085559328
 dialer-group 1
 ppp authentication chap pap
!
ip classless
ip route 0.0.0.0 0.0.0.0 172.18.38.40
dialer-list 1 protocol ip permit
!
line con 0
line 1
 password cisco
 login
 password cisco
 login
!
end
```

Large-Scale Dial-In Configuration Using Virtual Profiles

The following example is used on each central site stack member shown in Figure 119. This configuration is for a large-scale dial-in scenario.

```
aaa new-model
aaa authentication login default none
aaa authentication ppp default radius
aaa authentication ppp admin local
aaa authorization network radius
isdn switch-type primary-5ess
!
interface Serial0:23
 no ip address
 no ip mroute-cache
 no cdp enable
 ppp authentication chap
!
tacacs-server host 172.18.203.45
virtual-profile aaa
```

The following example configures an entry running on a RADIUS security server, which is queried by each central site stack member when a call comes in. This entry includes the virtual profile configuration information for remote users dialing in to the central site stack solution.

In this example, virtual profiles are configured by both virtual templates and AAA configuration. John and Rick can dial in from anywhere and have their same keepalive settings and their own IP addresses.

The remaining attribute-value pair settings are not used by virtual profiles. They are the network-protocol access lists and route filters used by AAA-based per-user configuration.

In the AAA configuration cisco-avpair lines, "\n" is used to indicate the start of a new Cisco IOS command line.

```
john Password = "welcome"
     User-Service-Type = Framed-User,
     Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=keepalive 75\nip address 100.100.100.100
255.255.255.0",
        cisco-avpair = "ip:rte-fltr-out#0=router igrp 60",
        cisco-avpair = "ip:rte-fltr-out#3=deny 171.0.0.0 0.255.255.255",
        cisco-avpair = "ip:rte-fltr-out#4=deny 172.0.0.0 0.255.255.255",
        cisco-avpair = "ip:rte-fltr-out#5=permit any"
rick Password = "emoclew"
     User-Service-Type = Framed-User,
     Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=keepalive 100\nip address 200.200.200.200
255.255.255.0",
        cisco-avpair = "ip:inacl#3=permit ip any any precedence immediate",
        cisco-avpair = "ip:inacl#4=deny igrp 0.0.1.2 255.255.0.0 any",
        cisco-avpair = "ip:outacl#2=permit ip any any precedence immediate",
        cisco-avpair = "ip:outacl#3=deny igrp 0.0.9.10 255.255.0.0 any"
```

Telecommuters Dialing In to a Mixed Protocol Environment

The scenario in this section describes how to provide remote access to employees who dial in to a mixed protocol enterprise network. The sample configurations provided in this section assume that enterprise telecommuters are dialing in with modems or terminal adapters from outside the LAN at headquarters.

The following sections are provided:

Description

Sometimes an enterprise conducts its daily business operations across internal mixed protocol environments. (See Figure 121 and Table 47.) For example, an enterprise might deploy an IP base across the entire intranet while still allowing file sharing with other protocols such as AppleTalk and AppleTalk Remote Access (ARA).

Figure 121 Large Enterprise with a Multiprotocol Network

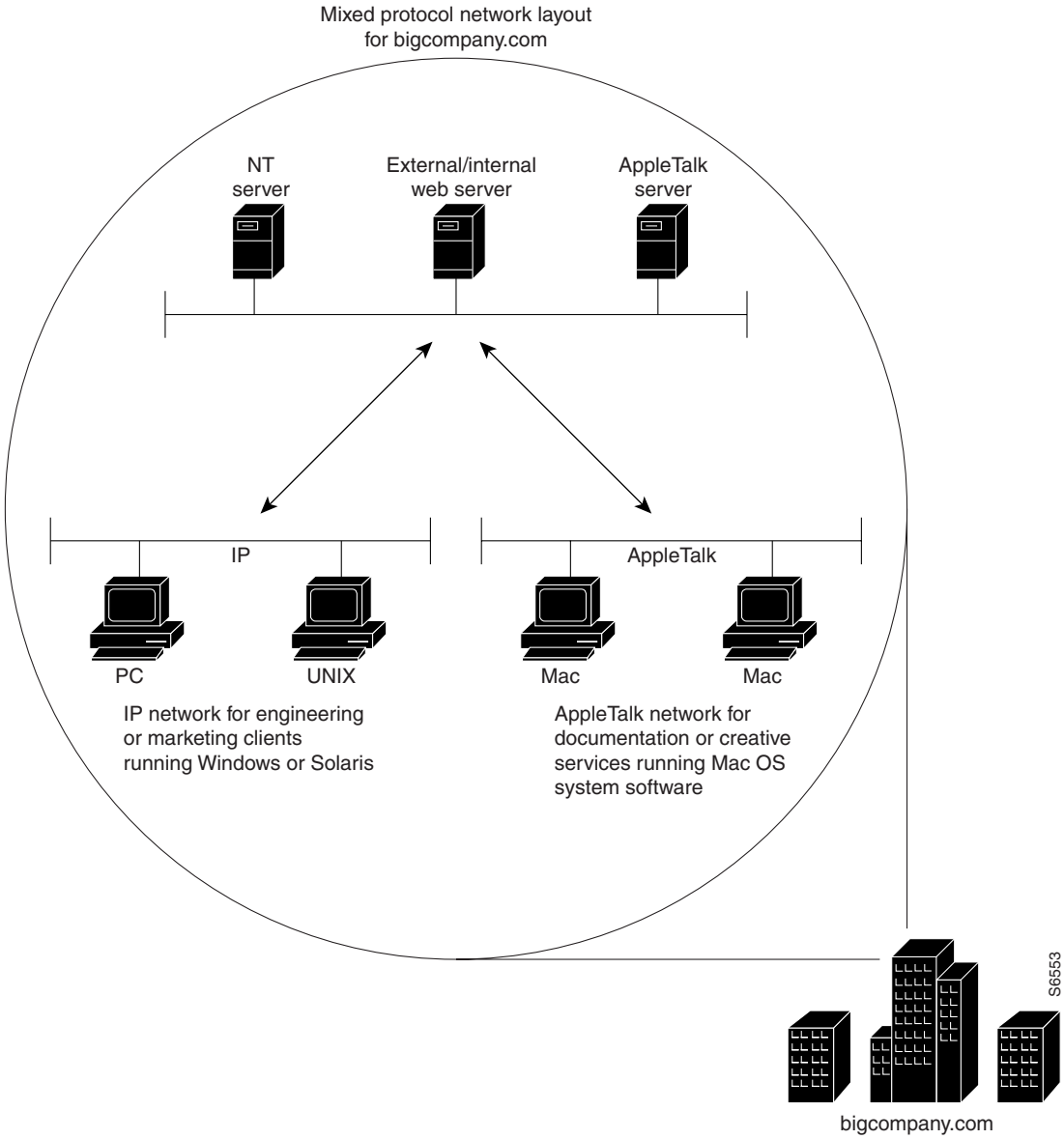


Table 47 Typical Mixed Protocol Environment

Applications Running on the Network Server	Remote or Local Client Applications	Protocol Used to Support the Network	Internal Supporting Department
Windows NT	Windows 95 or Windows 3.1 running on PCs	IP	Marketing, human resources, engineering, and customer support
UNIX	SunOS or Solaris running on a UNIX-based workstation or NCD	IP	Engineering and customer support

Table 47 Typical Mixed Protocol Environment (continued)

Applications Running on the Network Server	Remote or Local Client Applications	Protocol Used to Support the Network	Internal Supporting Department
AppleTalk	Mac OS System Software 7.5 running on Macintosh computers	AppleTalk	Documentation and creative services
NetWare	Novell NetWare client software	IPX	Marketing, and human resources, engineering, customer support

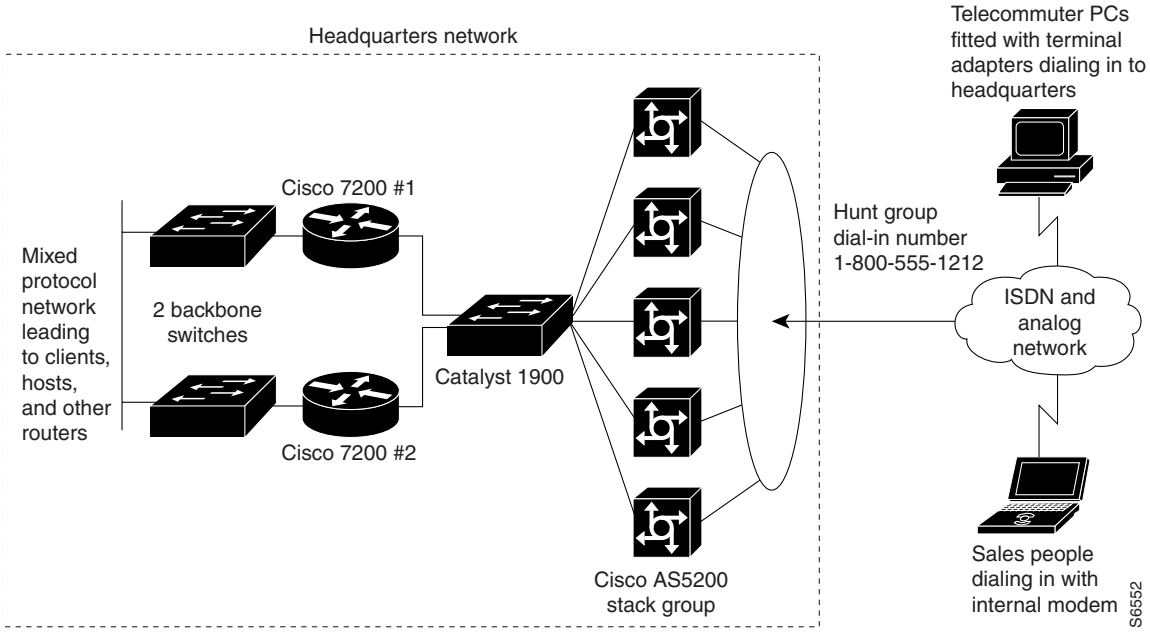
Enterprise Network Topology

Figure 122 shows a sample enterprise network, which supports 10,000 registered token card holders. Some registered users might use their access privileges each day, while others might use their access privileges very infrequently, such as only on business trips. The dial-in access provisioned for outsiders, such as partners or vendors, is supported separately in a firewalled setup.

Five Cisco AS5300 access servers are positioned to provide 250 dial-in ports for incoming modem calls. A Catalyst 1900 is used as a standalone switch to provide Ethernet switching between the Cisco AS5300 access servers and the 100BASET interfaces on the backbone routers. Two Cisco 7200 series routers are used to reduce the processing workload on the access servers and provide access to the company's backbone. If the Cisco 7200 series routers were not used in the network solution, the Cisco AS5300 access servers could not update routing tables, especially if 20 to 30 additional routers existed on the company's backbone. Two additional backbone switches are used to provide access to the company network.

Note Depending on your networking needs, the Cisco 7200 series routers could be substituted by one or more Cisco 3640 series routers. Additionally, the Cisco AS5300 access servers could be replaced by Cisco 3640 routers loaded with MICA digital modem cards.

Figure 122 Sample Enterprise Network Topology



If you are setting up dial-in access for remote terminal adapters, the settings configured on the terminal adapters must match the setting on the access server or router. Depending on your business application, terminal adapters can operate in many different modes. (See Table 48.)

Table 48 Options for Terminal Adapter Settings

Terminal Adapter Mode	Comments
Synchronous PPP	We recommend you use this mode for most terminal adapter scenarios. By default, Cisco access servers and routers have synchronous PPP enabled. Therefore, additional configuration is not required on the router or access server.
V.120	Use this mode for asynchronous to synchronous communication, which can be used to tunnel character mode sessions over synchronous ISDN. We recommend you use this mode with midrange routers, such as the Cisco 4500 series router.
V.110	Use this modem for setting up cellular modem access.

Mixed Protocol Dial-In Scenarios

The examples in the following sections are intended to run on each network device featured in Figure 122, which allows remote users to dial in to a mixed protocol environment:

- Cisco 7200 #1 Backbone Router
- Cisco 7200 #2 Backbone Router
- Cisco AS5300 Universal Access Server

Note Be sure to include your own IP addresses, host names, and security passwords where appropriate.

Cisco 7200 #1 Backbone Router

The following configuration runs on the router labeled Cisco 7200 #1 in Figure 122. Fast Ethernet interface 0/0 connects to the corporate backbone switch. Fast Ethernet interface 1/0 connects to the Catalyst 1900 switch, which in turn connects to the Cisco AS5300 access servers.

```
version xx.x
no service udp-small-servers
no service tcp-small-servers
!
hostname bbone-dial1
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
!
username admin password cisco
!
boot system flash slot0:
enable secret <password>
appletalk routing
ipx routing
!
interface FastEthernet0/0
 ip address 10.0.1.52 255.255.255.192
 appletalk cable-range 1000-1000
 appletalk zone Networking Infrastructure
 ipx network 1000
!
interface FastEthernet1/0
 ip address 10.1.1.2 255.255.255.224
 no ip redirects
 appletalk cable-range 7650-7650 7650.1
 appletalk zone Dial-Up Net
 ipx network 7650
!
 standby ip 10.1.1.1
 standby priority 101
 standby preempt
!
router eigrp 109
 redistribute static
 network 10.0.0.0
 no auto-summary
!
ip classless
ip http server
no logging console
!
ip route 10.1.2.0 255.255.255.192 10.1.1.10
!
line con 0
login authentication console
!
line vty 0 4
 login authentication default
end
```

Cisco 7200 #2 Backbone Router

The following configuration runs on the router labeled Cisco 7200 #2 in Figure 122. Fast Ethernet interface 0/0 connects to the corporate backbone switch. Fast Ethernet interface 1/0 connects to the Catalyst 1900 switch, which in turn connects to the Cisco AS5300 access servers.

```
version xx.x
no service udp-small-servers
no service tcp-small-servers
!
hostname bbone-dial2
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
!
username admin password cisco
!
boot system flash slot0:
enable secret <password>
appletalk routing
ipx routing
!
interface FastEthernet0/0
 ip address 10.0.1.116 255.255.255.192
 appletalk cable-range 1001-1001
 appletalk zone Networking Infrastructure
 ipx network 1001
!
interface FastEthernet1/0
 ip address 10.1.1.3 255.255.255.224
 no ip redirects
 appletalk cable-range 7650-7650 7650.2
 appletalk zone Dial-Up Net
 ipx network 7650
!
 standby ip 10.1.1.1
!
router eigrp 109
 redistribute static
 network 10.0.0.0
 no auto-summary
!
ip classless
ip http server
no logging console
!
ip route 10.1.2.0 255.255.255.192 10.1.1.10
!
line con 0
login authentication console
!
line vty 0 4
 login authentication console
!
end
```

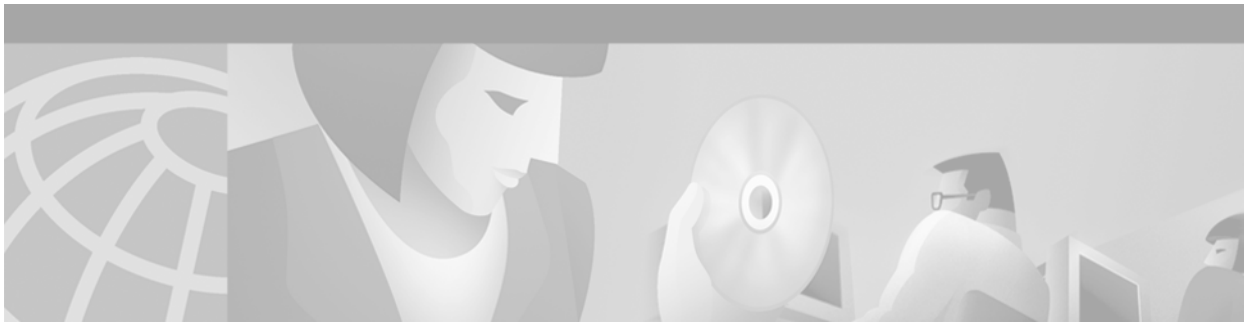
Cisco AS5300 Universal Access Server

The following configuration runs on each Cisco AS5300 access server in the stack group shown in Figure 122:

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
appletalk routing
ipx routing
appletalk virtual net 7651 Dial-Up Net
arap network 7652 Dial-Up Net
!
hostname NAS
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin local
aaa authentication ppp default local
aaa authentication ppp dialin if-needed local
aaa authentication arap default auth-guest local
enable secret cisco
!
username admin password cisco
username pcuser1 password mypass
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface loopback 0
 ip address 10.1.2.0 255.255.255.192
 ipx network 7651
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 appletalk cable-range 7650
 appletalk zone Dial-Up-Net
 ipx network 7650
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
```

```
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Group-Async1
 ip unnumbered Ethernet0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 appletalk client-mode
 ipx ppp-client
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
!
interface Dialer0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 ipx ppp-client
 appletalk client-mode
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
ip local pool dialin_pool 10.1.2.1 10.1.2.62
ip default-gateway 10.1.1.1
ip classless
ip route 0.0.0.0 0.0.0.0 10.1.1.1
!
dialer-list 1 protocol ip permit
!
async-bootp dns-server 10.1.0.40 10.1.0.170
async-bootp nbns-server 10.0.235.228 10.0.235.229
!
xremote buffersize 72000
xremote tftp host 10.0.2.74
!
```

```
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 autoselect arap
 arap enable
 arap authentication default
 arap timelimit 240
 arap warningtime 15
 login authentication dialin
 modem DialIn
 terminal-type dialup
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```



Telco and ISP Dial Scenarios and Configurations

This chapter provides sample hardware and software configurations for specific dial scenarios used by telcos, Internet service providers (ISPs), regional Bell operating companies (RBOCs), inter-exchange carriers (IXCs), and other service providers. Each configuration in this chapter is designed to enable IP network traffic with basic security authentication.

The following scenarios are described:

- Scenario 1—Small- to Medium-Scale POPs
- Scenario 2—Large-Scale POPs
- Scenario 3—PPP Calls over X.25 Networks

Note In all of these scenarios, you can replace the Cisco AS5200 access server with Cisco AS5300 or Cisco AS5800 access server. This hardware exchange provides higher call density performance and increases the number of PRI interfaces and modem ports on each chassis.

Small- to Medium-Scale POPs

Many small-to-medium-sized ISPs configure one or two access servers to provide dial-in access for their customers. Many of these dial-in customers use individual remote PCs that are not connected to LANs. Using the Windows 95 dialup software, remote clients initiate analog or digital connections using modems or home office ISDN BRI terminal adapters.

This section provides three types of single user dial-in scenarios for service providers:

- Individual Remote PCs Using Analog Modems
- Individual PCs Using ISDN Terminal Adapters
- Mixture of ISDN and Analog Modem Calls

Note Be sure to include your own IP addresses, host names, and security passwords where appropriate. The following sample configurations assume that the dial-in clients are individual PCs running PPP, connecting to an IP network, and requiring only basic security authentication.

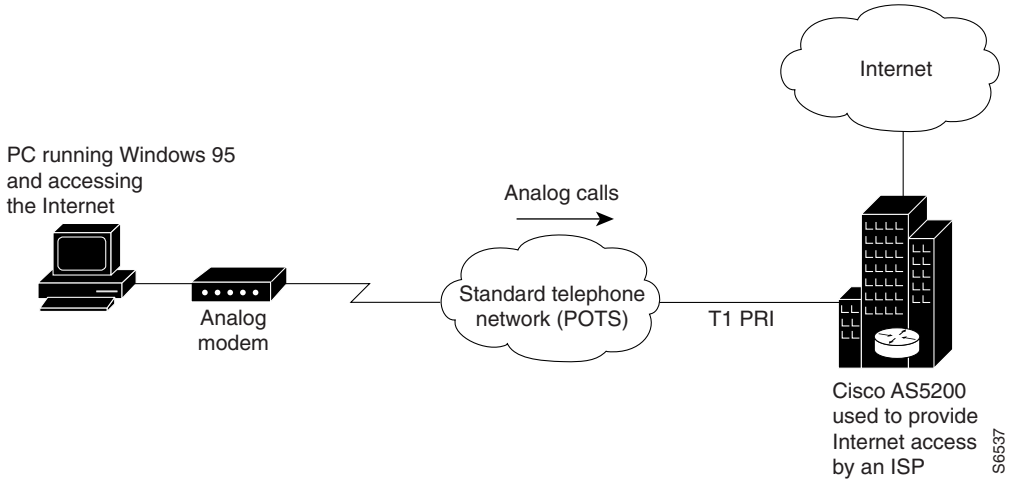
Individual Remote PCs Using Analog Modems

ISPs can configure a single Cisco access servers to receive analog calls from remote PCs connected to modems, as shown in Figure 123. The point of presence (POP) at the ISP central site could also be a Cisco 2511 access server connected to external modems.

Network Topology

Figure 123 shows a small-scale dial-in scenario using modems.

Figure 123 Remote PC Using an Analog Modem to Dial In to a Cisco Access Server



Running Configuration for ISDN PRI

The following sample configuration runs on the Cisco access server, as shown in Figure 123, which enables remote analog users to dial in:

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login console enable
aaa authentication login vty tacacs+
aaa authentication login dialin tacacs+
aaa authentication ppp default tacacs+
aaa authentication ppp dialin if-needed tacacs+
enable secret cisco
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
!
interface Serial1:23
 no ip address
 isdn incoming-voice modem
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
!
line aux 0
 login authentication console
```

```
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Some service providers use a remote TACACS+ or RADIUS security server in this dial-in scenario. The following example shows a TACACS+ entry that appears in the configuration file of a remote security server:

```
user = PCuser1 {
            login = cleartext "dialpass1"
            chap = cleartext "dialpass1"
            service = ppp protocol = ip {
                        addr-pool = dialin_pool
            }
            service = exec {
                        autocmd = "ppp negotiate"
            }
}


user = PCuser2 {
            login = cleartext "dialpass2"
            chap = cleartext "dialpass2"
            service = ppp protocol = ip {
                        addr-pool = dialin_pool
            }
            service = exec {
                        autocmd = "ppp negotiate"
            }
}


user = PCuser3 {
            login = cleartext "dialpass3"
            chap = cleartext "dialpass3"
            service = ppp protocol = ip {
                        addr-pool = dialin_pool
            }
            service = exec {
                        autocmd = "ppp negotiate"
            }
}
```

Running Configuration for Robbed-Bit Signaling

The following example shows a single Cisco access server configured to support remote client PCs dialing in with analog modems over traditional T1 lines. Digital ISDN calls do not transmit across these older types of channelized lines. The configuration assumes that the client can dial in and connect to the router in either terminal emulation mode (text only) or PPP packet mode.

Note The following configuration works only for analog modem calls. It includes no serial D-channel configuration (Serial 0:23 and Serial 1:23).

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
```

```
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login console enable
aaa authentication login vty tacacs+
aaa authentication login dialin tacacs+
aaa authentication ppp default tacacs+
aaa authentication ppp dialin if-needed tacacs+
enable secret cisco
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 cas-group 0 timeslots 1-24 type e&m-fgb
!
controller T1 1
 framing esf
 clock source line secondary
 linecode b8zs
 cas-group 0 timeslots 1-24 type e&m-fgb
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
!
```

```
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

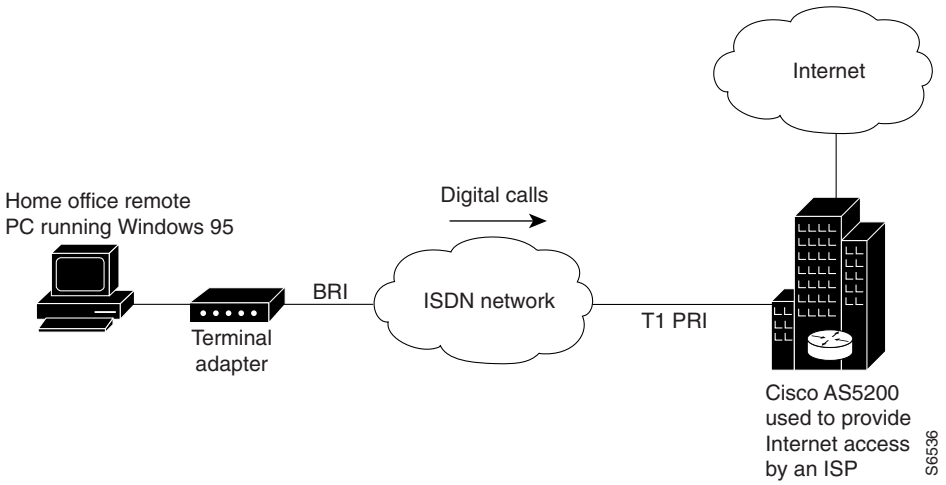
Individual PCs Using ISDN Terminal Adapters

ISPs can configure a single Cisco access server to receive digital multilink calls from remote PCs connected to terminal adapters, as shown in Figure 124. The POP at the central site of the ISP can be any Cisco router that supports ISDN PRI, such as the Cisco 4700-M router loaded with a channelized T1 PRI network module.

Network Topology

Figure 124 shows a small-scale dial-in scenario using terminal adapters.

Figure 124 Remote PC Using a Terminal Adapter to Dial In to a Cisco Access Server



To configure one Cisco access server to accept both incoming ISDN and analog calls from individual terminal adapters and modems, see the section "Mixture of ISDN and Analog Modem Calls" later in this chapter.

Terminal Adapter Configuration Example

The following example configures a Cisco access server to enable PCs fitted with internal or external terminal adapters to dial in to an IP network. The terminal adapter configuration is set up for asynchronous-to-synchronous PPP conversion. In some cases, PPP authentication must be set up for the Password Authentication Protocol (PAP). Some terminal adapters support only PAP authentication.

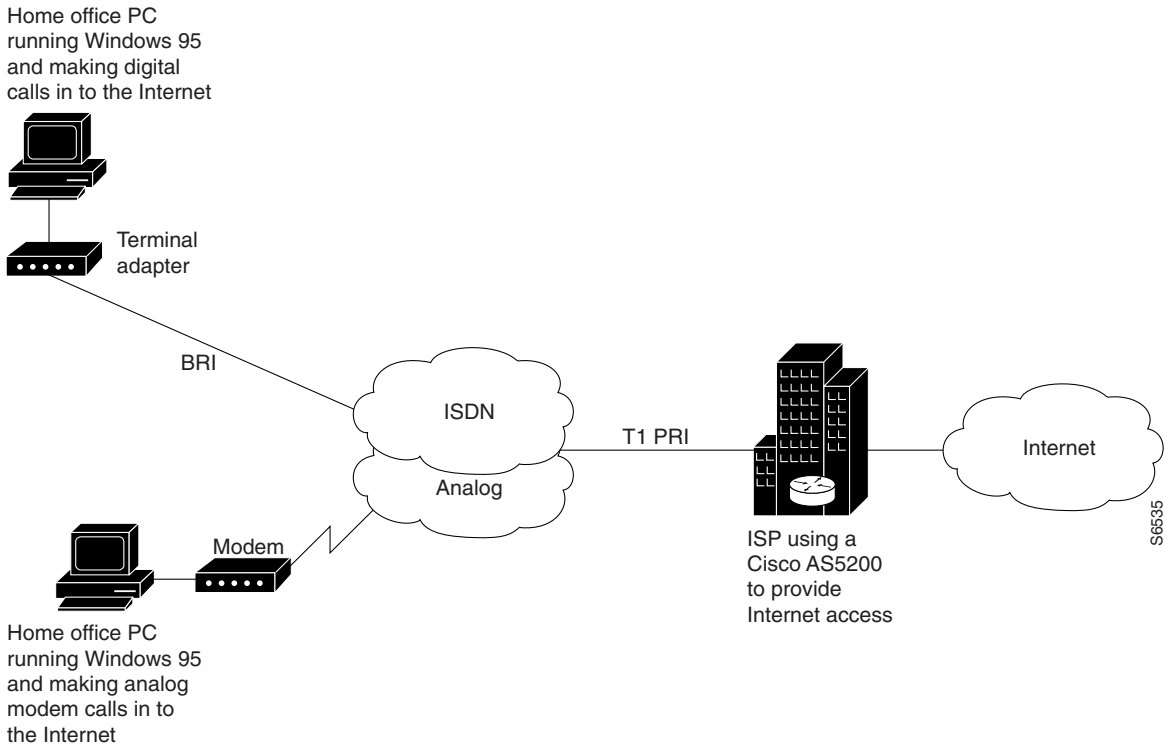
```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login console enable
aaa authentication login vty tacacs+
aaa authentication login dialin tacacs+
aaa authentication ppp default tacacs+
aaa authentication ppp dialin if-needed tacacs+
enable secret cisco
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
```

```
interface Serial1:23
 no ip address
 encapsulation ppp
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Mixture of ISDN and Analog Modem Calls

ISPs can configure a single Cisco access server to receive calls from a mixture of remote PCs connected to terminal adapters and modems, as shown in Figure 125.

Figure 125 Remote PCs Making Digital Calls and Analog Calls to a Cisco Access Server



Combination of Modem and ISDN Dial-In Configuration Example

The following example shows a combination of the modem and ISDN dial-in configurations. Using the bearer capability information element in the call setup packet, the incoming calls are labeled as data or voice. After the calls enter the access server, they are routed either to the serial configuration or to the modems and group asynchronous configuration.

Note This configuration assumes that only individual remote PCs are dialing in; no remote routers are dialing in. For a remote router dial-in configuration, see the chapter "Enterprise Dial Scenarios and Configurations" in this publication.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
```

```
aaa new-model
aaa authentication login console enable
aaa authentication login vty tacacs+
aaa authentication login dialin tacacs+
aaa authentication ppp default tacacs+
aaa authentication ppp dialin if-needed tacacs+
enable secret cisco
!
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
```

```
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
end
```

Large-Scale POPs

This section describes how to set up a stack of access servers for a large-scale dial solution and includes the following sections:

- Scaling Considerations
- How Stacking Works
- Stack Group of Access Servers Using MMP with an Offload Processor Examples

Scaling Considerations

Because of the significant increase in demand for Internet access, large POPs are required by many Telcos and ISPs. Internet access configurations can be set up to enable users who dial in with individual computers to make mixed ISDN multilink or modem connections using a stack of Cisco access servers that run Multichassis Multilink PPP (MMP).

You must consider scalability and call density issues when designing a large-scale dial-in POP. Because access servers have physical limitations, such as how many dial-in users can be supported on one device, you should consider the conditions and recommendations described in Table 49.

Table 49 Recommended Configurations for Different Remote Access Needs

Dial-in Demand You Need to Support	Recommended Configuration
PCs dialing in, 75 to 90 percent modem calls, 10 to 25 percent ISDN calls (terminal adapters or routers), and support for fewer than 96 (T1) to 116 (E1) simultaneous dial-in connections.	Two Cisco access servers configured for IP, basic security, MMP, L2F, and no offload server.
PCs dialing in, less than 50 percent modem calls, more than 50 percent ISDN calls (terminal adapters or routers), dial-in only, and 250 or more simultaneous links into the offload server.	Three or more Cisco access servers configured for IP, remote security, MMP, and L2F. Each Cisco access server is configured to offload its segmentation and reassembly of the multilink sessions onto an offload server, such as a Cisco 7202 or Cisco 4700 router.

Note Depending on the size of your POP requirement, you can replace the Cisco AS5200 access server with a Cisco AS5300, Cisco AS5800, or Cisco AccessPath. This hardware exchange provides higher call density performance and increases the number of ISDN PRI ports, channelized ports, and modem ports on each chassis.

How Stacking Works

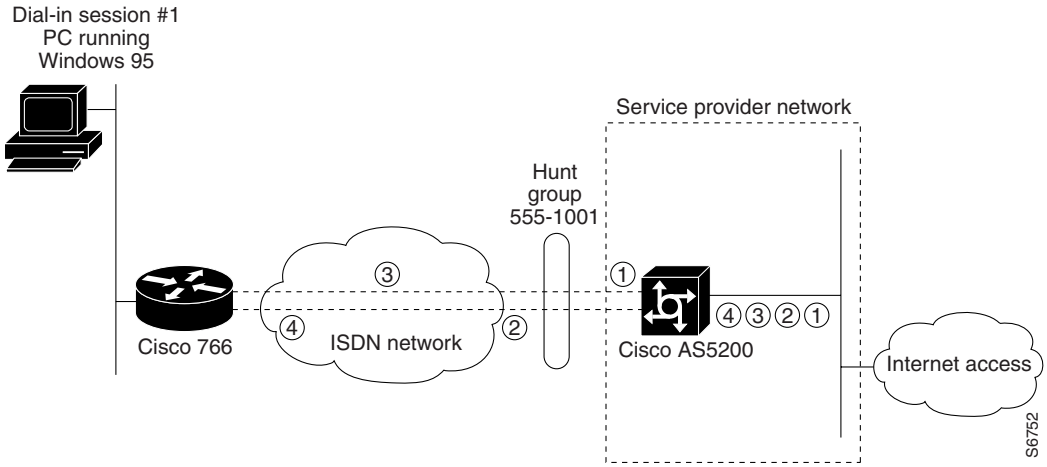
Before you install and configure a stack of access servers, you should understand the basic concepts described in the following sections and how they work together in a large-scale dial-in solution:

- A Typical Multilink PPP Session
- Using Multichassis Multilink PPP
- Setting Up an Offload Server
- Using the Stack Group Bidding Protocol
- Using L2F

A Typical Multilink PPP Session

A basic multilink session is an ISDN connection between two routing devices, such as a Cisco 766 router and a Cisco AS5200 access server. Figure 126 shows a remote PC connecting to a Cisco 766 ISDN router, which in turn opens two B-channel connections at 128 kbps across an ISDN network. The Multilink PPP (MLP) session is brought up. The Cisco 766 router sends four packets across the network to the Cisco AS5200, which in turn reassembles the packets back into the correct order and sends them out the LAN port to the Internet.

Figure 126 A Typical Multilink PPP Session

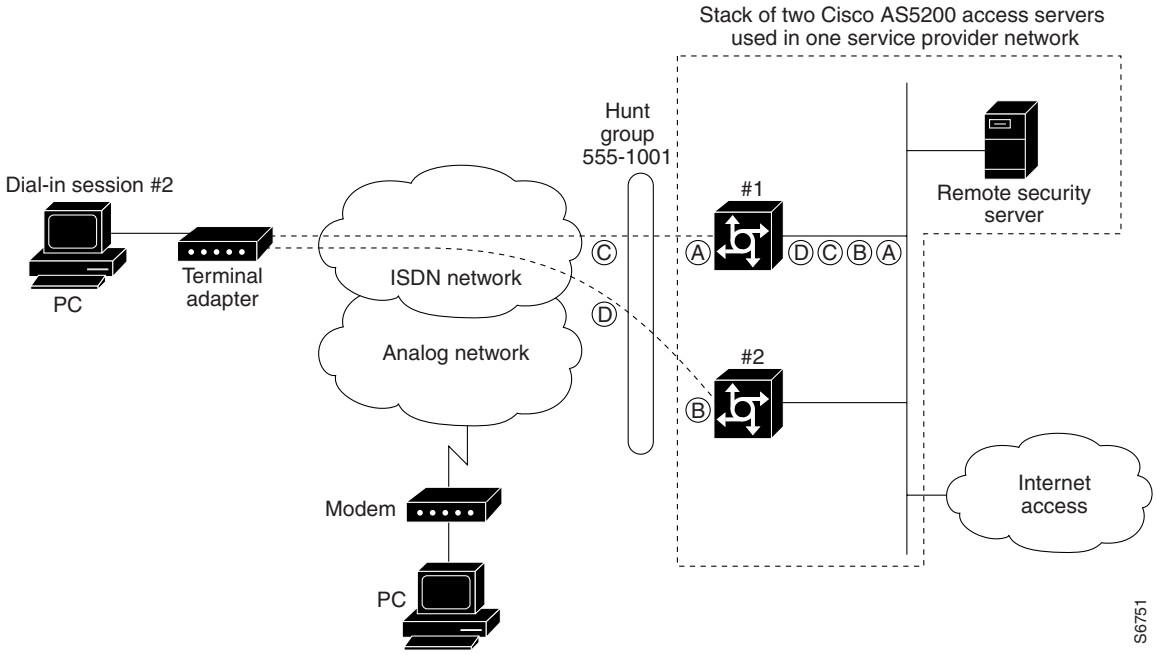


Using Multichassis Multilink PPP

The dial solution becomes more complex when the scenario is scaled to include multiple multilink calls connecting across multiple chassis. Figure 127 shows a terminal adapter making a call in to the Cisco AS5200, labeled #1. However, only one of the access server's 48 B channels is available to accept the call. The other channels are busy with calls. As a result, one of the terminal adapter's two B channels is redirected to device #2. At this point, a multilink multichassis session is shared between two Cisco AS5200s that belong to the same stack group. Packet fragments A and C go to device #1. Packet fragments B and D go to device #2.

Because device #1 is the first access server to receive a packet and establish a link, this access server creates a virtual interface and becomes the bundle master. The bundle master takes ownership of the MLP session with the remote device. The Multichassis Multilink PPP (MMP) protocol forwards the second link from device #2 to the bundle master, which in turn bundles the two B channels together and provides 128 kbps to the end user. Layer 2 Forwarding (L2F) is the mechanism that device #2 uses to forward all packet fragments received from the terminal adapter to device #1. In this way, all packets and calls virtually appear to terminate at device #1.

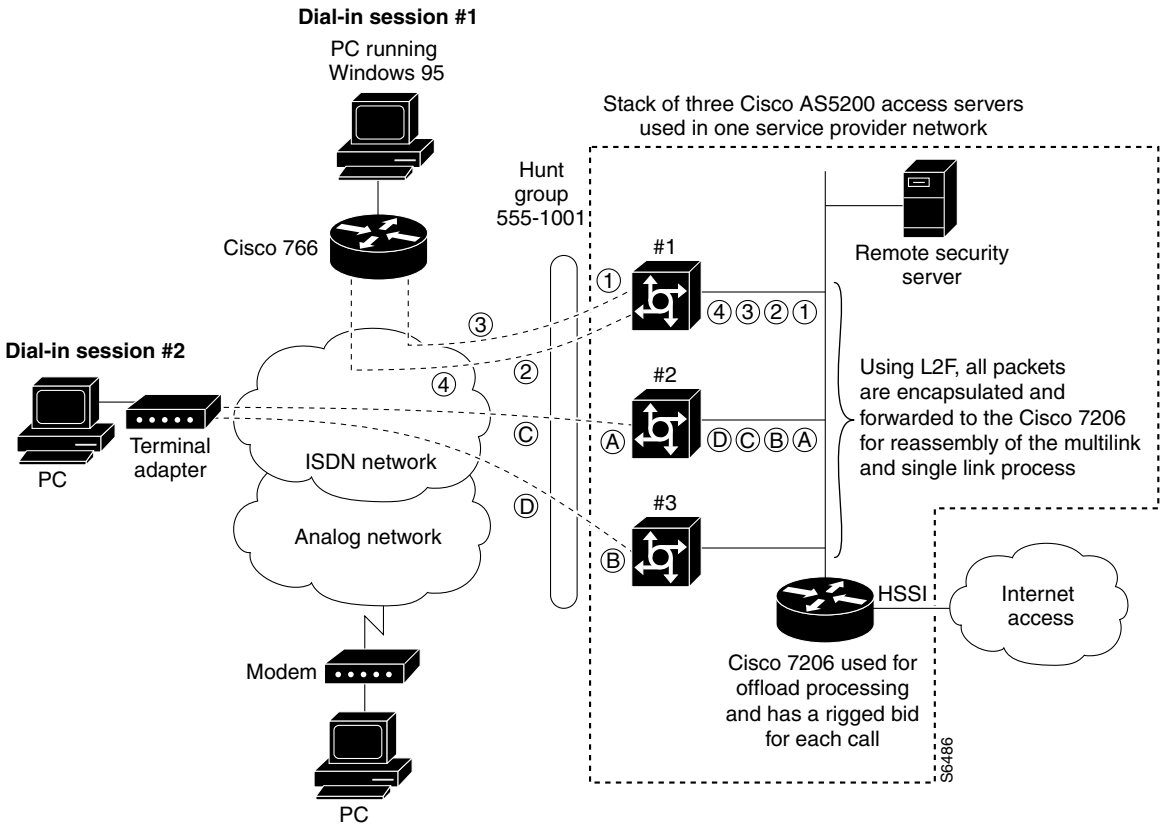
Figure 127 A Stack Group of Access Servers Using MMP Without an Offload Processor



Setting Up an Offload Server

Because MMP is a processor-intensive application, you might need to offload the processing or segmentation and reassembly from the Cisco access servers to a router with a more powerful CPU, such as the Cisco 4700-M or Cisco 7206. We recommend that you include an offload server for dial-in solutions that support more than 50 percent ISDN calls or more than 10 multilink sessions per Cisco access server. (See Figure 128.)

Figure 128 A Stack Group of Access Servers Using MMP with an Offload Processor



Using the Stack Group Bidding Protocol

The Stack Group Bidding Protocol (SGBP) is a critical component used in multichassis multilink sessions. SGBP unites each Cisco access server in a virtual stack, which enables the access servers to become virtually tied together. Each independent stack member communicates with the other members and determines which devices' CPU should be in charge of running the multilink session and packet reassembly—the duty of the bundle master. The goal of SGBP is to find a common place to forward the links and ensure that this destination has enough CPU power to perform the segmentation and packet reassembly. (See Figure 128.)

When SGBP in configured on each Cisco access server, each access server sends out a query to each stack group member stating, for example, "I have a call coming in from walt@options.com. What is your bid for this user?" Each access server then consults the following default bidding criteria and answers the query accordingly:

- Do I have an existing call or link for the user walt@options.com? If I do, then bid very high to get this second link in to me.
- If I do not have an existing call for walt@options.com, then bid a value that is proportional to how much CPU power I have available.
- How busy am I supporting other users?

Note An offload server will always serve as the bundle master by bidding a higher value than the other devices.

Using L2F

L2F is a critical component used in multichassis multilink sessions. If an access server is not in charge of a multilink session, the access server encapsulates the fragmented PPP frames and forwards them to the bundle master using L2F. The master device receives the calls, not through the dial port (such as a dual T1/PRI card), but through the LAN or Ethernet port. L2F simply tunnels packet fragments to the device that owns the multilink session for the call. If you include an offload server in your dial-in scenario, it creates all the virtual interfaces, owns all the multilink sessions, and reassembles all the fragmented packets received by L2F via the other stackgroup members. (Refer to Figure 128.)

Stack Group of Access Servers Using MMP with an Offload Processor Examples

The following sections provide examples for the devices shown in Figure 128:

- Cisco Access Server #1
- Cisco Access Server #2
- Cisco Access Server #3
- Cisco 7206 as Offload Server
- RADIUS Remote Security Examples

Note Be sure to include your own IP addresses, host names, and security passwords where appropriate.

Cisco Access Server #1

The following configuration runs on the Cisco access server labeled #1 in Figure 128:

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname AS5200-1
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin radius
aaa authentication ppp default local
aaa authentication ppp dialin if-needed radius
aaa authorization exec local radius
aaa authorization network radius
aaa accounting network start-stop radius
aaa accounting exec start-stop radius
```

```
enable secret cisco
!
username admin password cisco
username MYSTACK password STACK-SECRET
sgbp group MYSTACK
sgbp member AS5200-2 10.1.1.12
sgbp member AS5200-3 10.1.1.13
sgbp member 7200 10.1.1.14
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.62 255.255.255.192
!
interface Ethernet0
 ip address 10.1.1.11 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.192
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
```

```
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
radius-server host 10.1.1.23 auth-port 1645 acct-port 1646
radius-server host 10.1.1.24 auth-port 1645 acct-port 1646
radius-server key cisco
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Cisco Access Server #2

The following configuration runs on the Cisco access server labeled #2 shown in Figure 128:

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname AS5200-2
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin radius
aaa authentication ppp default local
aaa authentication ppp dialin if-needed radius
aaa authorization exec local radius
```

```
aaa authorization network radius
aaa accounting network start-stop radius
aaa accounting exec start-stop radius
enable secret cisco
!
username admin password cisco
username MYSTACK password STACK-SECRET
sgbp group MYSTACK
sgbp member AS5200-1 10.1.1.11
sgbp member AS5200-3 10.1.1.13
sgbp member 7200 10.1.1.14
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.126 255.255.255.192
!
interface Ethernet0
 ip address 10.1.1.12 255.255.255.0
 ip summary address eigrp 10 10.1.2.64 255.255.255.192
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
```

```
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0..0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.65 10.1.2.114
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
radius-server host 10.1.1.23 auth-port 1645 acct-port 1646
radius-server host 10.1.1.24 auth-port 1645 acct-port 1646
radius-server key cisco
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Cisco Access Server #3

The following configuration runs on the Cisco access server labeled #3 in Figure 128:

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname AS5200-3
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin radius
```

```
aaa authentication ppp default local
aaa authentication ppp dialin if-needed radius
aaa authorization exec local radius
aaa authorization network radius
aaa accounting network start-stop radius
aaa accounting exec start-stop radius
enable secret cisco
!
username admin password cisco
username MYSTACK password STACK-SECRET
sgbp group MYSTACK
sgbp member AS5200-1 10.1.1.11
sgbp member AS5200-2 10.1.1.12
sgbp member 7200 10.1.1.14
async-bootp dns-server 10.1.3.1 10.1.3.2
isdn switch-type primary-5ess
!
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
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback0
 ip address 10.1.2.190 255.255.255.192
!
interface Ethernet0
 ip address 10.1.1.13 255.255.255.0
 ip summary address eigrp 10 10.1.2.128 255.255.255.192
!
interface Serial0
 no ip address
 shutdown
!
interface Serial1
 no ip address
 shutdown
!
interface Serial0:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
interface Serial1:23
 no ip address
 encapsulation ppp
 isdn incoming-voice modem
 dialer rotary-group 0
 dialer-group 1
 no fair-queue
 no cdp enable
!
```

```
interface Group-Async1
 ip unnumbered Loopback0
 encapsulation ppp
 async mode interactive
 peer default ip address pool dialin_pool
 no cdp enable
 ppp authentication chap pap dialin
 group-range 1 48
!
interface Dialer0
 ip unnumbered Loopback0
 no ip mroute-cache
 encapsulation ppp
 peer default ip address pool dialin_pool
 dialer in-band
 dialer-group 1
 no fair-queue
 no cdp enable
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.129 10.1.2.178
ip default-gateway 10.1.1.1
ip classless
!
dialer-list 1 protocol ip permit
radius-server host 10.1.1.23 auth-port 1645 acct-port 1646
radius-server host 10.1.1.24 auth-port 1645 acct-port 1646
radius-server key cisco
!
line con 0
 login authentication console
line 1 48
 autoselect ppp
 autoselect during-login
 login authentication dialin
 modem DialIn
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
 transport input telnet rlogin
!
end
```

Cisco 7206 as Offload Server

The following configuration runs on the Cisco 7206 router shown in Figure 128:

Note Any Cisco router that has a powerful CPU can be used as an offload server, such as a Cisco 4500-M, 4700-M, or 3640. However, the router must be configured to handle the necessary processing overhead demanded by each stack member.

```
version xx.x
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname 7200
!
aaa new-model
aaa authentication login default local
aaa authentication login console enable
aaa authentication login vty local
aaa authentication login dialin radius
aaa authentication ppp default local
aaa authentication ppp dialin if-needed radius
aaa authorization exec local radius
aaa authorization network radius
aaa accounting network start-stop radius
aaa accounting exec start-stop radius
enable secret cisco
!
username MYSTACK password STACK-SECRET
username admin password cisco
multilink virtual-template 1
sgbp group MYSTACK
sgbp member AS5200-1 10.1.1.11
sgbp member AS5200-2 10.1.1.12
sgbp member AS5200-3 10.1.1.13
sgbp seed-bid offload
async-bootp dns-server 10.1.3.1 10.1.3.2
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.192
!
interface Ethernet2/0
 ip address 10.1.1.14 255.255.255.0
 ip summary address eigrp 10 10.1.2.192 255.255.255.192
!
interface Ethernet2/1
 no ip address
 shutdown
!
interface Ethernet2/2
 no ip address
 shutdown
!
interface Ethernet2/3
 no ip address
 shutdown
!
```

```
interface Virtual-Template1
 ip unnumbered Loopback0
 no ip mroute-cache
 peer default ip address pool dialin_pool
 ppp authentication chap pap dialin
 ppp multilink
!
router eigrp 10
 network 10.0.0.0
 passive-interface Virtual-Template1
 no auto-summary
!
ip local pool dialin_pool 10.1.2.193 10.1.2.242
ip default-gateway 10.1.1.1
ip classless
!
radius-server host 10.1.1.23 auth-port 1645 acct-port 1646
radius-server host 10.1.1.24 auth-port 1645 acct-port 1646
radius-server key cisco
!
line con 0
 login authentication console
line aux 0
 login authentication console
line vty 0 4
 login authentication vty
!
end
```

RADIUS Remote Security Examples

The RADIUS examples in the following sections use the Internet Engineering Task Force (IETF) syntax for the attributes:

- User Setup for PPP
- User Setup for PPP and Static IP Address
- Enabling Router Dial-In
- User Setup for SLIP
- User Setup for SLIP and Static IP Address
- Using Telnet to connect to a UNIX Host
- Automatic rlogin to UNIX Host

Depending on how the dictionary is set up, the syntax for these configurations might differ between versions of RADIUS daemons.

Note You must have the **async dynamic address** command enabled on the network access server if you use Framed-IP-Address to statically assign IP addresses.

User Setup for PPP

The following example shows a user setup for PPP. The user's IP address comes from the configured default IP address that is set up on the interface (which could be a specific default IP address, a pointer to a local pool of addresses, or a pointer to a Dynamic Host Configuration Protocol (DHCP) server). The special address that signals the default address is 255.255.255.254.

```
pppme Password = "cisco"
        CHAP-Password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = PPP,
        Framed-IP-Address = 255.255.255.254
```

User Setup for PPP and Static IP Address

The following example shows a user setup for PPP and a static IP address that stays with the user across all connections. Make sure that your router is set up to support this configuration, especially for large or multiple POPs.

```
staticallypppme Password = "cisco"
        CHAP-Password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = PPP,
        Framed-IP-Address = 10.1.1.1
```

Enabling Router Dial-In

The following example supports a router dialing in, which requires that a static IP address and a remote Ethernet interface be added to the network access server's routing table. The router's WAN port is assigned the address 1.1.1.2. The remote Ethernet interface is 2.1.1.0 with a class C mask. Be sure your routing table can support this requirement. You might need to redistribute the static route with a dynamic routing protocol.

```
routeme Password = "cisco"
        CHAP-Password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = PPP,
        Framed-IP-Address = 10.1.1.1
        Framed-Route = "10.2.1.0/24 10.1.1.2"
```

User Setup for SLIP

The following example shows a user setup for SLIP. Remote users are assigned to the default address on the interface.

```
slipme Password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = SLIP,
        Framed-IP-Address = 255.255.255.254
```

User Setup for SLIP and Static IP Address

The following example shows a user setup for SLIP and a static IP address that stays with the user across all connections. Make sure that your routing is set up to support this configuration, especially for large or multiple POPs.

```
staticallyslipme Password = "cisco"
        Service-Type = Framed,
        Framed-Protocol = SLIP,
        Framed-IP-Address = 10.1.1.13
```

Using Telnet to connect to a UNIX Host

The following example automatically uses Telnet to connect the user to a UNIX host. This configuration is useful for registering new users, providing basic UNIX shell services, or providing a guest account.

```
telnetme Password = "cisco"
        Service-Type = Login,
        Login-Service = Telnet,
        Login-IP-Host = 10.2.1.1
```

Automatic rlogin to UNIX Host

The following example automatically uses rlogin to connect the user to a UNIX host:

```
rloginme Password = "cisco"
        Service-Type = Login,
        Login-Service = Rlogin,
        Login-IP-Host =10.3.1.2
```

If you want to prevent a second password prompt from being brought up, you must have the following two commands enabled on the router or access server:

- **rlogin trusted-remoteuser-source local**
- **rlogin trusted-localuser-source radius**

PPP Calls over X.25 Networks

Remote PCs stationed in X.25 packet assembler-disassembler (PAD) networks can access the Internet by dialing in to Cisco routers, which support PPP. By positioning a Cisco router at the corner of an X.25 network, ISPs and telcos can provide Internet and PPP access to PAD users. All remote PAD users that dial in to X.25 networks dial in to one Cisco router that allows PPP connections. Although connection performance is not optimal, these X.25-to-PPP calls use installed bases of X.25 equipment and cost less to operate than connecting over the standard telephone network.

Note This dial-in scenario can also be used as an enterprise solution. In this case, an enterprise consults with a third-party service provider that allows enterprises to leverage existing X.25 enterprise equipment to provide connections back into enterprise environments.

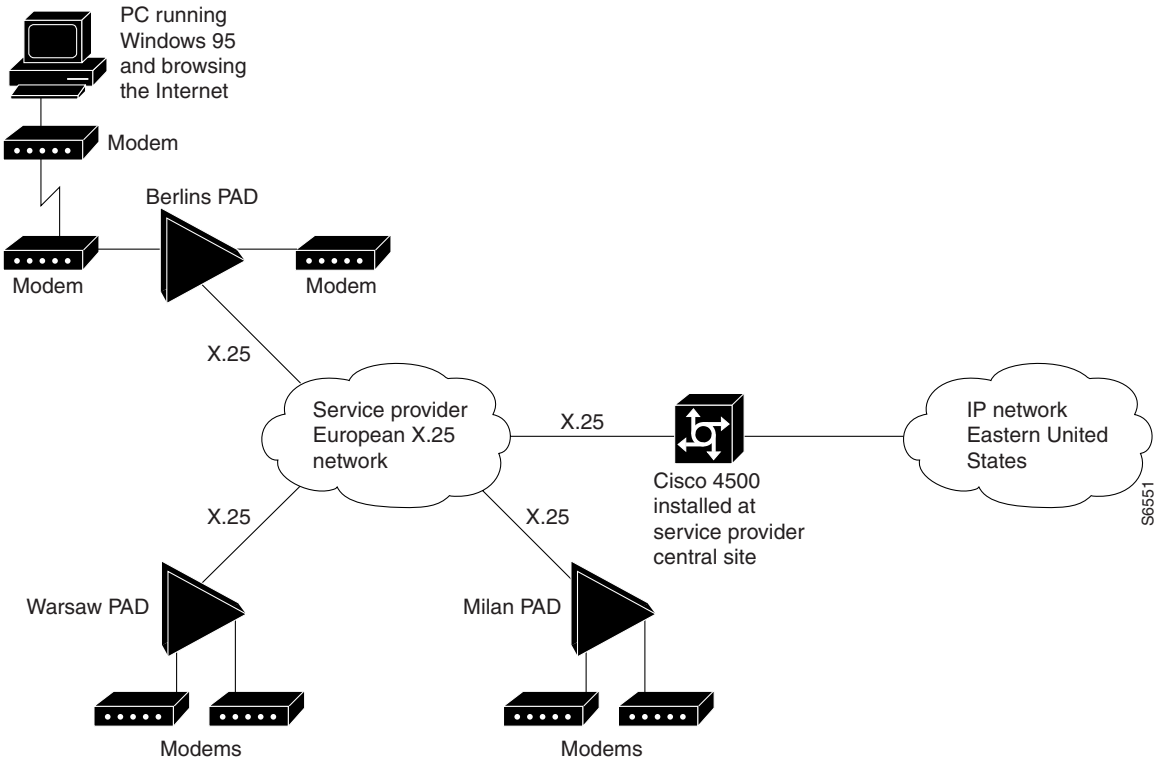
Overview

Many cities throughout the world have large installed bases of PCs that interface with older modems, PADs, and X.25 networks. These remote PCs or terminals dial in to PADs and make X.25 PAD calls or terminal connections to mainframe computers or other devices, which run the X.25 protocol. Unfortunately, the user interface is only a regular text-based screen in character mode (as opposed to packet mode). Therefore, many ISPs and telcos that have large investments in X.25 networks are upgrading their outdated equipment and creating separate networks for PPP connections. Because this upgrade process takes substantial time and money to complete, using a Cisco router to allow PPP connections over an X.25 network is a good interim solution for a dead-end dial case.

Remote PC Browsing Network Topology

Figure 129 shows a remote PC browsing the Internet through an X.25 PAD call and a Cisco 4500 router. This X.25 network is owned by an ISP or telco that is heavily invested in X.25 equipment, that is currently upgrading its outdated equipment, and that is creating separate networks for PPP connections. In this topology, the Cisco 4500 router performs protocol translation between the protocols X.25 and PPP. The router is configured to accept an incoming X.25 PAD call, run and unpack PPP packets over the call, and enable the remote PC to function as if it were on the IP network.

Figure 129 Remote PC Browsing the Internet Through an X.25 PAD Call and a Cisco 4500 Router



For more information about configuring protocol translation, see the chapter "Configuring Protocol Translation and Virtual Asynchronous Devices" in the *Cisco IOS Terminal Services Configuration Guide*.

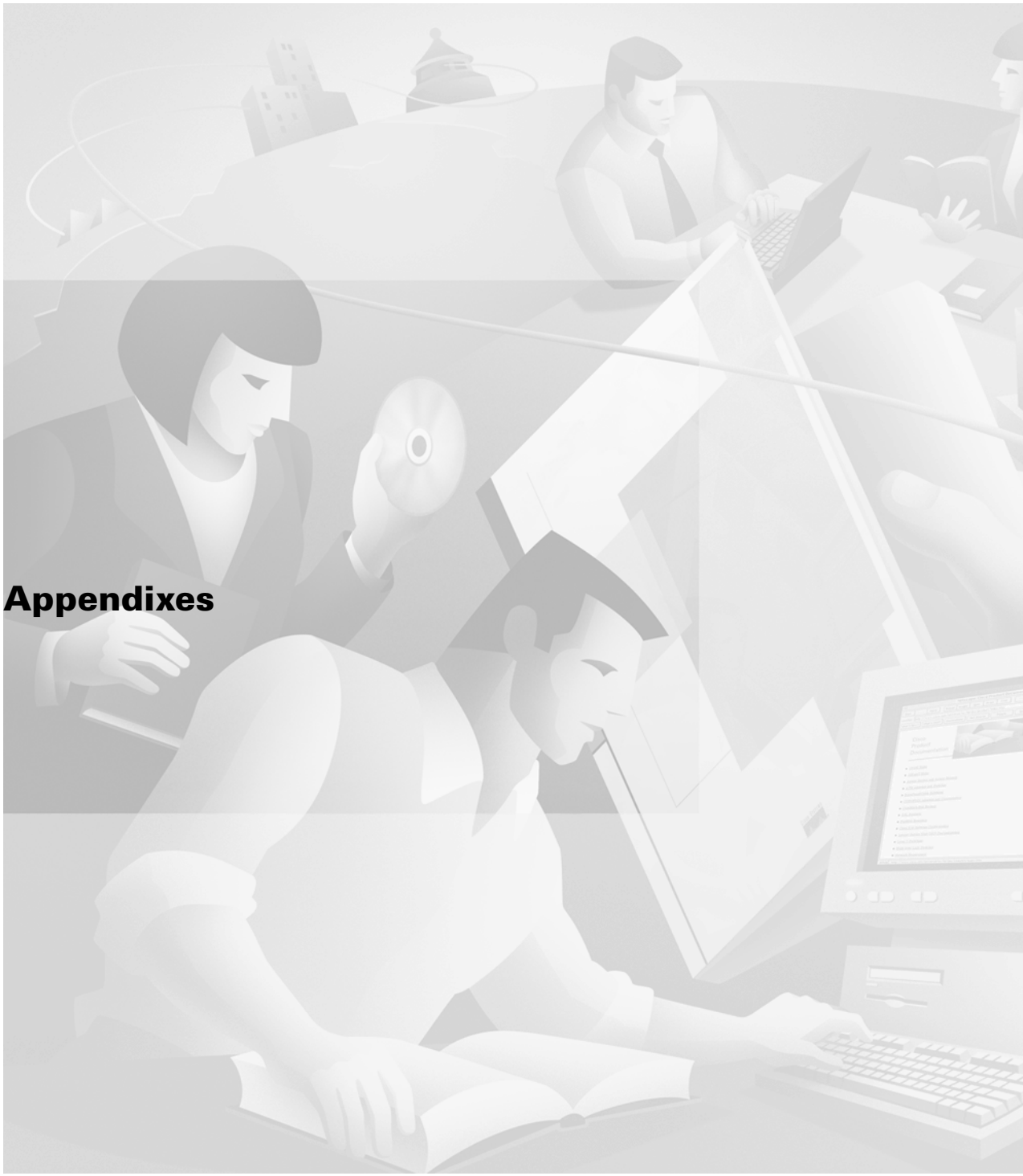
Protocol Translation Configuration Example

In the following example, PAD callers that dial 4085551234 receive a router prompt. PAD callers that dial 4085555123401 start PPP and pick up an address from the IP pool called dialin_pool. These addresses are "borrowed" from the Ethernet interface on the Cisco 4500 router. Additionally, a loopback interface network can be created and the X.25 addresses can be set. However, a routing protocol must be run to advertise the loopback interface network if this method is used.

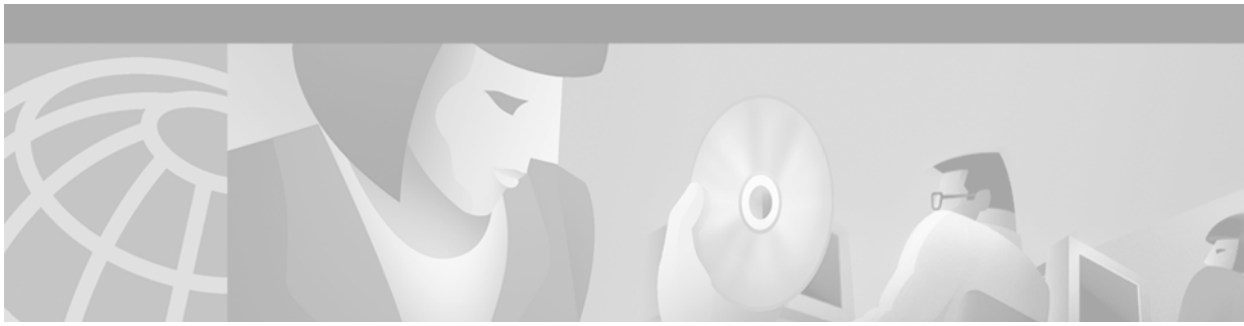
Note Be sure to include your own IP addresses, host names, and security passwords where appropriate in the following examples.

```
service timestamps debug datetime msec
service timestamps log datetime msec
service password-encryption
no service udp-small-servers
no service tcp-small-servers
!
hostname NAS
!
aaa new-model
aaa authentication login console enable
aaa authentication login vty tacacs+
aaa authentication login dialin tacacs+
aaa authentication ppp default tacacs+
aaa authentication ppp dialin if-needed tacacs+
enable secret cisco
!
async-bootp dns-server 10.1.3.1 10.1.3.2
!
vty-async
vty-async ppp authentication chap pap
!
interface Loopback0
 ip address 10.1.2.254 255.255.255.0
!
interface Ethernet0
 ip address 10.1.1.10 255.255.255.0
 ip summary address eigrp 10 10.1.2.0 255.255.255.0
!
interface Serial0
 no ip address
 encapsulation x25
 x25 address 4085551234
 x25 accept-reverse
 x25 default pad
!
router eigrp 10
 network 10.0.0.0
 passive-interface Dialer0
 no auto-summary
!
ip local pool dialin_pool 10.1.2.1 10.1.2.50
ip default-gateway 10.1.1.1
!
ip classless
!
translate x25 4085555123401 ppp ip-pool scope-name dialin_pool
!
dialer-list 1 protocol ip permit
!
```

```
line con 0
 login authentication console
line aux 0
 login authentication console
line vty 0 150
 login authentication vty
 transport input telnet rlogin
!
end
```



Appendixes



Modem Initialization Strings

This appendix provides tables that contain modem initialization strings and sample modem initialization scripts. Table 50 lists required settings, and error compression (EC) and compression settings for specific modem types. Use this information to create your modem scripts. Table 51 lists information for setting AUX ports. SeeTable 52 for a legend of symbols used in these two tables. Sample scripts follow the tables.

For information about configuring lines to support modems, see the chapters in the part "Modem and Dial Shelf Configuration and Management" in this publication.

Table 50 Required Settings and EC/Compression Settings

Settings Required for All Modems					**Settings for EC/Compression**					
Modem	**FD**	**AA**	**CD**	**DTR**	**RTS/CTS Flow**	**LOCK DTE Speed**	**Best Error**	**Best Comp**	**No Error**	**No Comp**
Codex 3260	&F	S0=1	&C1	&D3	*FL3	*SC1	*SM3	*DC1	*SM1	*DC0
USR Courier USR Sportster	&F	S0=1	&C1	&D3	&H1&R 2	&B1	&M4	&K1	&M0	&K0
Global Village Teleport Gold	&F	S0=1	&C1	&D3	\Q3	\J0	\N7	%C1	\N0	%C0
Telebit T1600/T3000/ WB	&F1	S0=1	&C1	&D3	S58=2 S68=2	S51=6	S180=2 S181=1	S190=1	S180=0 S181=1	S190=0
Telebit T2500 (ECM)	&F	S0=1	&C1	&D3	S58=2 S68=2	S51=6	S95=2	S98=1 S96=1	S95=0	S98=0 S96=0
Telebit Trailblazer	&F	S0=1	&C1							
AT&T Paradyne Dataport	&F	S0=1	&C1	&D3	\Q3	--->	\N7	%C1	\N0	%C0
Hayes modems Accura/ Optima	&F	S0=1	&C1	&D3	&K3	&Q6	&Q5	&Q9	&Q6	<---
Microcom QX4232 series	&F	S0=1	&C1	&D3	\Q3	\J0	\N6	%C1	\N0	%C0

Table 50 Required Settings and EC/Compression Settings (continued)

Settings Required for All Modems					Settings for EC/Compression					
Modem	**FD**	**AA**	**CD**	**DTR**	**RTS/CTS Flow**	**LOCK DTE Speed**	**Best Error**	**Best Comp**	**No Error**	**No Comp**
Motorola UDS FastTalk II	&F	S0=1	&C1	&D3	\Q3	\J0	\N6	%C1	\N0	%C0
Multitech MT1432 MT932	&F	S0=1	&C1	&D3	&E4	$BA0	&E1	&E15	&E0	&E14
Digicom Scout Plus	&F	S0=1	&C1	&D3	*F3	*S1	*E9	<---	*E0	<---
Digicom SoftModem	&F	S0=1	&C1	&D3	&K3	--->	\N5	%C1	\N0	%C0
Viva 14.4/9642c	&F	S0=1	&C1	&D3	&K3	--->	\N3	%M3	\N0	%M0
ZyXel U-1496E	&F	S0=1	&C1	&D3	&H3	&B1	&K4	<---	&K0	<---
Supra V.32bis/28.8	&F	S0=1	&C1	&D3	&K3	--->	\N3	%C1	\N0	%C0
ZOOM 14.4	&F	S0=1	&C1	&D3	&K3	--->	\N3	%C2	\N0	%C0
Intel External	&F	S0=1	&C1	&D3	\Q3	\J0	\N3	%C1"H 3	\N0	%C0
Practical Peripherals	&F	S0=1	&C1	&D3	&K3	--->	&Q5	&Q9	&Q6	<---

Table 51 AUX and Platform Specific Settings

Modem	Settings for Use with AUX Port		Other Settings		Comments
	No Echo	No Res	CAB-MDCE	Write Memory	
Codex 3260	E0	Q1	&S1	&W	
USR Courieræ USR Sportster	E0	Q1	*NA*	&W	
Global Village Teleport Gold	E0	Q1	*NA*	&W	
Telebit T1600/T3000/ WB	E0	Q1	&S4	&W	All Telebit modems need to have the speed set explicitly. These examples use 38400 bps. Using what Telebit calls "UNATTENDED ANSWER MODE" is the best place to start a dial in only modem.
Telebit T2500 (ECM)	E0	Q1	&S1	&W	
Telebit Trailblazer	E0	Q1	*NA*	&W	Use "ENHANCED COMMAND MODE" on the T2500.
AT&T Paradyne Dataport	E0	Q1	*NA*	&W	Almost all Microcom modems have similar configuration parameters.
Hayes modems Accura/ Optima	E0	Q1	*NA*	&W	
Microcom QX4232 series	E0	Q1	*NA*	&W	
Motorola UDS FastTalk II	E0	Q1	*NA*	&W	
Multitech MT1432 MT932	E0	Q1	&S1	&W	
Digicom Scout Plus	E0	Q2	&B2	&W	
Digicom SoftModem	E0	Q1	&S1	&W	
Viva 14.4/9642c	E0	Q1	&S1	&W	
ZyXel U-1496E	E0	Q1	&S1	&W	Additional information on ftp.zyxel.com
Supra V.32*bis*/28.8	E0	Q1	&S1	&W	
ZOOM 14.4	E0	Q1	&S1	&W	

Table 51 AUX and Platform Specific Settings (continued)

Modem	Settings for Use with AUX Port		Other Settings		Comments
	No Echo	No Res	CAB-MDCE	Write Memory	
Intel External	E0	Q1	*NA*	&W	
Practical Peripherals	E0	Q1	*NA*	&W	Based on PC288LCD. May vary.

Table 52 contains a legend of symbols used in Table 50 and Table 51.

Table 52 Legend to Symbols Used in Modem Chart

Symbol	Meaning
NA	This option is not available on the noted modem.
-->	The command noted on the right will handle that function.
<--	The command noted on the left will handle that function.
AUX port	These parameters are only required for pre-9.21 AUX ports or any other port without modem control set.

Sample Modem Scripts

The following are several modem command strings that are appropriate for use with your access server or router. For use with the access server, **Speed=xxxxxx** is a suggested value only. Set the DTE speed of the modem to its maximum capability. By making a reverse Telnet connection in the EXEC mode to the port on the access server where the modem is connected, then sending an **at** command followed by a carriage return.

In the following example, the modem is attached to asynchronous interface 2 on the access server. The IP address indicated as the server-ip-address is the IP address of the Ethernet 0 interface. The administrator connects from the EXEC to asynchronous interface 2, which has its IP address assigned from Ethernet 0.

```
2511> telnet server-ip-address port-number
              192.156.154.42    2002
```

AST Premium Exec Internal Data/Fax (MNP 5)

```
Init=AT&F&C1&D3\G0\J0\N3\Q2S7=60S0=1&W
Speed=9600
```

ATi 9600etc/e (V.42*bis*)

```
Init=AT&FW2&B1&C1&D3&K3&Q6&U1S7=60S0=1&W
Speed=38400
```

AT&T Paradyne KeepInTouch Card Modem (V.42*bis*)

```
Init=AT&FX6&C1&D3\N7\Q2%C1S7=60S0=1&w
Speed=57600
```

AT&T ComSphere 3800 Series (V.42*bis*)

```
Init=AT&FX6&C1&D2\N5\Q2%C1"H3S7=60S0=1&W
Speed=57600
```

AT&T DataPort Fax Modem (V.42*bis*)

```
Init=AT&FX6&C1&D2\N7\Q2%C1S7=60S0=1&W
Speed=38400
```

Boca Modem 14.4K/V.32*bis* (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5%C1\N3S7=60S36=7S46=138S95=47S0=1&W
Speed=57600
```

CALPAK MXE-9600

```
Init=AT&F&C1&D3S7=60S0=1&W
Speed=9600
```

Cardinal 2450MNP (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2\V1%C1S7=60S0=1&w
Speed=9600
```

Cardinal 9650V32 (MNP)

```
Init=AT&F&B1&C1&D3&H1&I1&M6S7=60S0=1&W
```

Cardinal 9600V42 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5\N3%C1%M3S7=60S46=138S48=7S95=3S0=1&W
Speed=38400
```

Cardinal 14400 (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C1%M3S7=60S46=138S48=7S95=47S0=1&W
Speed=57600
```

COMPAQ SpeedPAQ 144 (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\J0\N3%C1S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=57600
```

Data Race RediMODEM V.32/V.32*bis*

```
Init=AT&F&C1&D3&K3&Q6\J0\N7\Q3\V2%C1S7=60 Speed=38400S0=1&W
```

Dell NX20 Modem/Fax (MNP)

```
Init=AT&F&C1&D3%C1\J0\N3\Q3\V1W2S7=60S0=1&W
Speed=9600
```

Digicom Systems (DSI) 9624LE/9624PC (MNP 5)

```
Init=AT&F&C1&D3*E1*F3*S1S7=60S0=1&W
```

Digicom Systems (DSI) 9624LE+ (V.42*bis*)

```
Init=AT&F&C1&D3*E9*F3*N6*S1S7=60S0=1&W
Speed=38400
```

Everex Evercom 24+ and 24E+ (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2\V1%C1S7=60S0=1&W
```

Everex EverFax 24/96 and 24/96E (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2\V1%C1S7=60S0=1&W
Speed=9600
```

Everex Evercom 96+ and 96E+ (V.42bis)

```
Init=AT&FW2&C1&D3\J0\N3\Q2\V2%C1S7=60S0=1&W
Speed=38400
```

Freedom Series V.32*bis* Data/FAX Modem

```
Init=AT&F&C1&D3&K3&Q6\J0\N7\Q3\V2%C1S7=60S0=1&W
Speed=38400
```

Gateway 2000 TelePath

```
Init=AT&FW2&C1&D3&K3&Q5\N3%C1S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

Gateway 2000 Nomad 9600 BPS Internal Modem

```
Init=AT&F&C1&D3%C1\J0\N3\Q2S7=60S0=1&W
Speed=38400
```

GVC SM-96V (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N6\Q2\V1S7=60S0=1&W
Speed=38400
```

GVC SM-144V (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N6\Q2\V1S7=60S0=1&W
Speed=57600
```

Hayes Smartmodem Optima 9600 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S46=138S48=7S95=47S0=1&W
Speed=38400
```

Hayes Smartmodem Optima 14400 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S46=138S48=7S95=47S0=1&W
Speed=57600
```

Hayes Optima 28800 (V.34)

```
Init=AT&FS0=1&C1&D3&K3&Q6&Q5&Q9&W
Speed=115200
```

Hayes V-series Smartmodem 9600/9600B (V.42)

```
Init=AT&F&C1&D3&K3&Q5S7=60S0=1&W
Speed=9600
```

Hayes V-series ULTRA Smartmodem 9600 (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S46=2S48=7S95=63S0=1&W
Speed=38400
```

Hayes V-series ULTRA Smartmodem 14400 (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S38=10S46=2S48=7S95=63S0=1&W
Speed=38400
```

Hayes ACCURA 24 EC (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S36=7S46=138S48=7S95=47S0=1&W
```

Hayes ACCURA 96 EC (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

Hayes ACCURA 144 EC (V.42*bis*)

```
Init=AT&FW2&C1&D3&K3&Q5S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=57600
```

Hayes ISDN System Adapter

```
Init=AT&FW1&C1&D3&K3&Q0S7=60S0=1&W
Speed=57600
```

IBM 7855 Modem Model 10 (MNP)

```
Init=AT&F&C1&D3\N3\Q2\V1%C1S7=60S0=1&W
```

IBM Data/Fax Modem PCMCIA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5%C3\N3S7=60S38=7S46=138S48=7S95=47S0=1&W
Speed=57600
```

Identity ID9632E

```
Init=AT&F&C1&D3S7=60S0=1&W
Speed=9600
```

Infotel V.42X (V.42*bis*)

```
Init=AT&F&C1&D3S7=30S36=7S0=1&W
Speed=9600
```

Infotel V.32 turbo (V.42*bis*)

```
Init=AT&FW1&C1&D3&K3&Q5S7=60S0=1&w
Speed=38400
```

Infotel 144I (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C1S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

Intel 9600 EX (V.42*bis*)

```
Init=AT&F&C1&D3\J0\N3\Q2\V2%C1"H3S7=60S0=1&W
Speed=38400
```

Intel 14400 EX (V.42*bis*)

```
Init=AT&F&C1&D3\J0\N3\Q2\V2%C1"H3S7=60S0=1&W
Speed=38400
```

Macronix MaxFax 9624LT-S

```
Init=AT&F&C1&D3&K3&Q9\J0\N3\Q3%C1S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=9600
```

Megahertz T3144 internal (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N3\Q2\V2S7=60S0=1&W
Speed=57600
```

Megahertz T324FM internal (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N3\Q2\V1S7=60S46=138S48=7S0=1&W
Speed=9600
```

Megahertz P2144 FAX/Modem (V.42*bis*)

```
Init=AT&F&C1&D3%C1\J0\N7\Q2\V2S7=60S0=1&W
Speed=38400
```

Megahertz T396FM internal (V.42*bis*)

```
Init=AT&FW2&C1&D3%C1\J0\N7\Q2\V2S7=60S0=1&W
Speed=38400
```

Megahertz CC3144 PCMCIA card modem (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5%C3\N3S7=60S38=7S46=138S48=7S95=47S0=1&W
Speed=57600
```

Microcom AX/9624c (MNP 5)

```
Init=AT&F&C1&D3\G0\J0\N3\Q2%C1S7=60S0=1&W
Speed=9600
```

Microcom AX/9600 Plus (MNP 5)

```
Init=AT&F&C1&D3\J0\N3\Q2S7=60S0=1&W
```

Microcom QX/V.32c (MNP 5)

```
Init=AT&F&C1&D3\J0%C3\N3\Q2S7=60S0=1&W
Speed=38400
```

Microcom QX/4232hs (V.42*bis*)

```
Init=AT&F&C1&D3\J0%C3\N3\Q2-K0\V2S7=60S0=1&W
Speed=38400
```

Microcom QX/4232*bis* (V.42*bis*)

```
Init=AT&F&C1&D3\J0%C3\N3\Q2-K0\V2W2S7=60S0=1&W
Speed=38400
```

Microcom Deskporte 28800 (V.34)

```
Init=AT&F&c1&q1E0S0=1&W
Speed=115200
```

Microcom MicroPorte 542 (V.42*bis*)

```
Init=AT&F&C1&D3&Q5S7=60S46=138S48=7S95=47S0=1&W
Speed=9600
```

Microcom MicroPorte 1042 (V.42*bis*)

```
Init=AT&F&C1&D3%C3\J0-M0\N6\Q2\V2S7=60S0=1&W
Speed=9600
```

Microcom MicroPorte 4232*bis* (V.42*bis*)

```
Init=AT&F&C1&D3%C3%G0\J0-M0\N6\Q2\V2S7=60S0=1&W
Speed=38400
```

Microcom DeskPorte FAST

```
Init=ATX4S7=60-M1\V4\N2L1S0=1&W
Speed=57600
```

Motorola/Codex 3220 (MNP)

```
Init=AT&F&C1&D3*DC1*FL3*MF0*SM3*XC2S7=60S0=1&W
```

Motorola/Codex 3220 Plus (V.42*bis*)

```
Init=AT&F&C1&D3*DC1*EC0*MF0*SM3*XC2S7=60S0=1&W
Speed=38400
```

Motorola/Codex 326X Series (V.42*bis*)

```
Init=AT&F&C1&D3*FL3*MF0*SM3*TT2*XC2S7=60S0=1&W
Speed=38400
```

MultiTech MultiModem V32EC (V.42*bis*)

```
Init=AT&FX4&C1&D3$BA0&E1&E4&E15#L0S7=60S0=1&W
Speed=38400
```

MultiTech MultiModem V32 (no MNP or V.42)

```
Init=AT&F&C1&D3S7=60S0=1&W
Speed=9600
```

MultiTech MultiModem 696E (MNP)

```
Init=AT&F&C1&D3$BA0&E1&E4&E15S7=60S0=1&W
```

MultiTech MultiModem II MT932 (V.42*bis*)

```
Init=AT&FX4&C1&D3$BA0&E1&E4&E15#L0S7=60S0=1&W
Speed=38400
```

MultiTech MultiModem II MT1432 (V.42*bis*)

```
Init=AT&FX4&C1&D3#A0$BA0&E1&E4&E15#L0S7=60S0=1&W
Speed=57600
```

NEC UltraLite 14.4 Data/Fax Modem (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q4\J0\N7\Q2W2%C1S7=60S0=1&W
Speed=38400
```

Practical Peripherals PC28800SA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=115200
```

Practical Peripherals PM9600SA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S46=138S48=7S7=60S0=1&W
Speed=38400
```

Practical Peripherals PM14400FX (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=57600
```

Practical Peripherals PM14400SA (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5S7=60S36=7S46=2S48=7S95=47S0=1&W
Speed=57600
```

Prometheus ProModem 9600 Plus (V.42)

```
Init=AT&F&C1&D3*E7*F3S7=60S0=1&W
```

Prometheus ProModem Ultima (V.42*bis*)

```
Init=AT&F&C1&D3*E9*F3*N6*S1S7=60S0=1&W
Speed=38400
```

Racal Datacomm ALM 3223 (V.42*bis*)

```
Init=AT&F&C1&D3\M0\N3\P2\Q1\V1S7=60S0=1&W
Speed=38400
```

Supra FAXModem V.32*bis* (V.42*bis*)

```
Init=AT&FN1W2&C1&D1&K3&Q5\N3%C1S7=60S36=7S48=7S95=45S0=1&W
Speed=57600
```

Telebit T1600 (V.42*bis*)

```
Init=AT&FX2&C1&D3&R3S7=60S51=6S58=0S59=15S68=2S180=2S190=1S0=1&W
Speed=38400
```

Telebit T2500 (V.42*bis*)

```
Init=AT~&FX2S7=60S51=5S52=2S66=1S68=2S97=1S98=3S106=1S131=1S0=1&W

Telebit T3000 (V.42bis)
Init=AT&FX2&C1&D3S51=6S59=7S68=2S7=60S0=1&W
Speed=38400
```

Telebit QBlazer (V.42*bis*)

```
Init=AT&FX2&C1&D3S59=7S68=2S7=60S0=1&W
Speed=38400
```

Texas Instruments V.32*bis* Internal Modem

```
Init=AT&F&C1&D3%C1\J0\N7\Q2\V2S7=60S0=1&W
Speed=38400
```

Toshiba T24/DF Internal

```
Init=AT&F&C1&D3\J0\N3\Q2%C1S7=60S36=7S46=138S48=7S0=1&W
Speed=9600
```

Universal Data Systems FasTalk V.32/42b (V.42*bis*)

```
Init=AT&F&C1&D3\J0\M0\N7\V1\Q2%C1S7=60S0=1&W
Speed=38400
```

Universal Data Systems V.32 (no MNP or V.42)

```
Init=AT&F&C1&D2S7=60S0=1&W
Speed=9600
```

Universal Data Systems V.3224 (MNP 4)

```
Init=AT&F&C1&D2\J0\N3\Q2S7=60S0=1&W
```

Universal Data Systems V.3225 (MNP 5)

```
Init=AT&F&C1&D2\J0\N3\Q2%C1S7=60S0=1&W
```

Universal Data Systems V.3227 (V.42*bis*)

```
Init=AT&F&C1&D2\J0\M0\N7\Q2%C1S7=60S0=1&W
Speed=38400
```

Universal Data Systems V.3229 (V.42*bis*)

```
Init=AT&F&C1&D3\J0\M0\N7\Q2%C1S7=60S0=1&W
Speed=38400
```

US Robotics Sportster 9600 (V.42*bis*)

```
Init=AT&FX4&A3&B1&D3&H1&I0&K1&M4S7=60S0=1&W
Speed=38400
```

US Robotics Sportster 14400 (V.42*bis*)

```
Init=AT&FX4&A3&B1&D3&H1&I0&K1&M4S7=60S0=1&W
Speed=57600
```

US Robotics Sportster 14400 (V.42*bis*) x

```
Init=AT&FX4&B1&C1&D2&H1&K1&M4E0X7Q0V1S0=1&W
Speed=57600
```

US Robotics Sportster 28800 (V.34)

```
Init=AT&FS0=1&C1&D2&H1&R2&N14&B1&W
Speed=115200
```

US Robotics Courier 28800 (V.34)

```
Init=AT&FS0=1&C1&D2&H1&R2&N14&B1&W
Speed=115200
```

US Robotics Courier V.32bis (V.42*bis*)

```
Init=AT&FX4&A3&C1&D2&M4&H1&K1&B1S0=1&W
Speed=38400
```

US Robotics Courier HST Dual Standard (V.42*bis*)

```
Init=AT&FB0X4&A3&C1&D2&M4&H1&K1&B1&R2&S1S0=1&W
Speed=115200
```

US Robotics Courier HST (V.42*bis*)

```
Init=AT&FB0X4&A3&C1&D2&M1&H1&K1&B1S0=1&W
Speed=115200
```

US Robotics WorldPort 2496 FAX/Data (V.42*bis*)

```
Init=AT&FX4&C1&D3%C1"H3\J0-J1\N3\Q2\V2S7=60S0=1&W
Speed=57600
```

US Robotics WorldPort 9696 FAX/Data (MNP 5)

```
Init=AT&FX4&C1&D3%C1\J0\N3\Q2\V2S7=60S0=1&W
```

US Robotics WorldPort 9600 (MNP 5)

```
Init=AT&FX4&C1&D3%C1\J0\N3\Q2\V2S7=60S0=1&W
```

US Robotics WorldPort 14400 (V.42*bis*)

```
Init=AT&FX4&A3&B1&C1&D3&H1&K1&M4S7=60S0=1&W
Speed=57600
```

Ven-Tel PCM 9600 Plus (MNP)

```
Init=AT&FB0&C1&D3\N3\Q3%B0%C1%F1S7=60S0=1&W
```

ViVa 9642e (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C3S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

ViVa 14.4/FAX (V.42*bis*)

```
Init=AT&F&C1&D3&K3&Q5\N3%C3S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

ZOOM V.32 turbo (V.42*bis*)

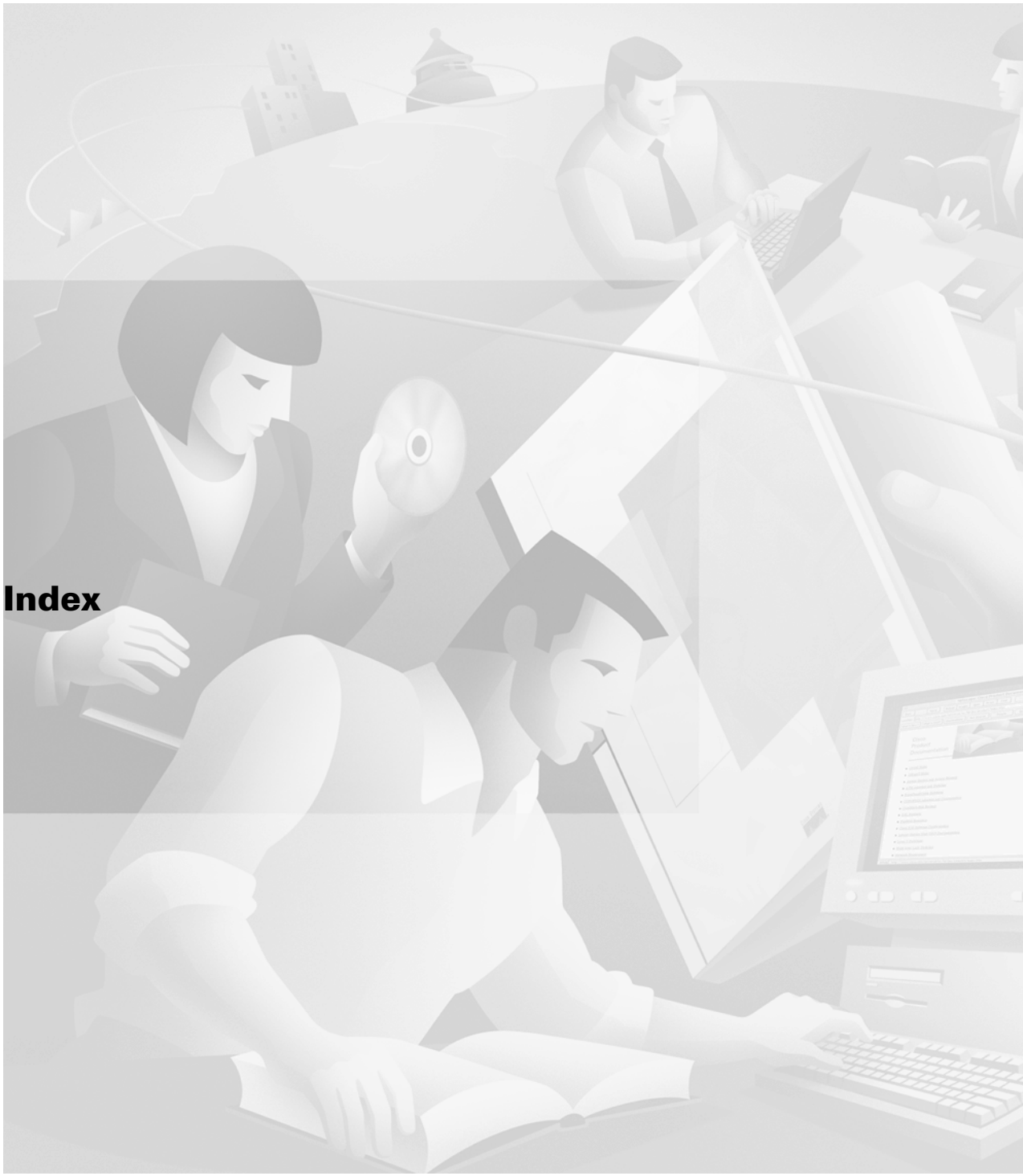
```
Init=AT&FW1&C1&D3&K3&Q5%C1\N3S7=60S36=7S46=138S48=7S95=47S0=1&W
Speed=38400
```

ZOOM V.32*bis* (V.42*bis*)

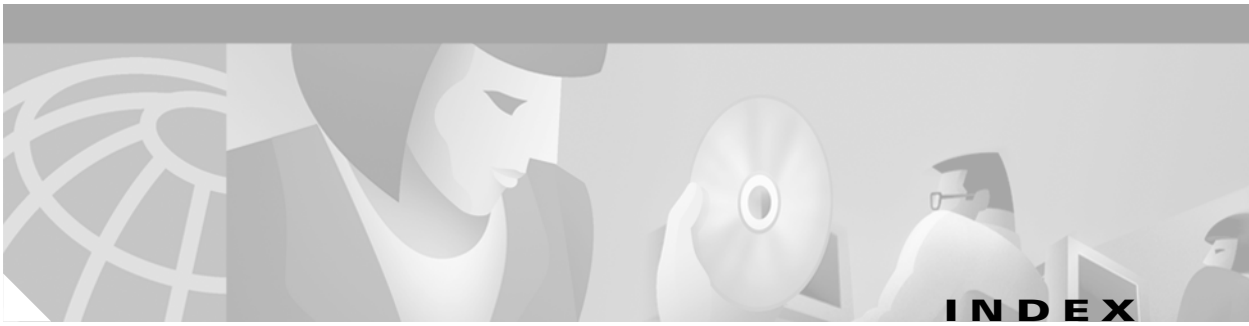
```
Init=AT&FW1&C1&D3&K3&Q9%C1\N3S7=60S36=7S95=47S0=1&W
Speed=38400
```

Zyxel U-1496 (V.42*bis*)

```
Init=AT&FX6&B1&C1&D2&N0&K4&H3S7=60S0=1&W
Speed=57600
```



Index



Symbols

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

X