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
- Asynchronous Serial Traffic over UDP Configuration Task List

See the "UDPTN Configuration Examples" 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.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

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.

Asynchronous Serial Traffic over UDP Configuration Task List

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 "UDPTN Configuration Examples" 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
```

UDPTN Configuration Examples

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 Signal and Line States

First Published: May 8, 2001
Last Updated: May 14, 2009

This chapter contains information on how to configure automatic dialing for modems and provides illustrations describing modem signal and line states.

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Modem Signal and Line State" section on page 16.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About Signal and Line State Diagrams

Signal and line state diagrams accompany some of the tasks in the following sections to illustrate how the modem control works. The following diagrams are described here:

EXEC and Daemon Creation on a Line with No Modem Control

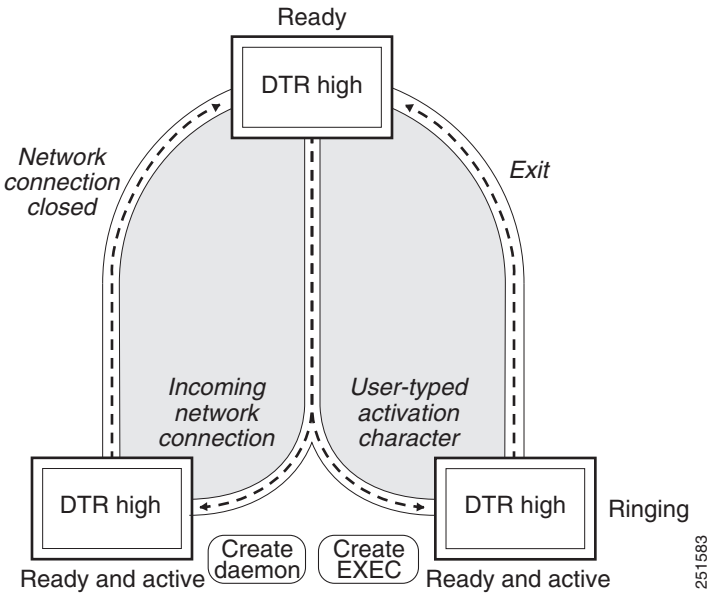
The diagrams show two processes:

- 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 1 through Figure 5.)

In the diagrams, the current signal state and the signal that 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 1 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.

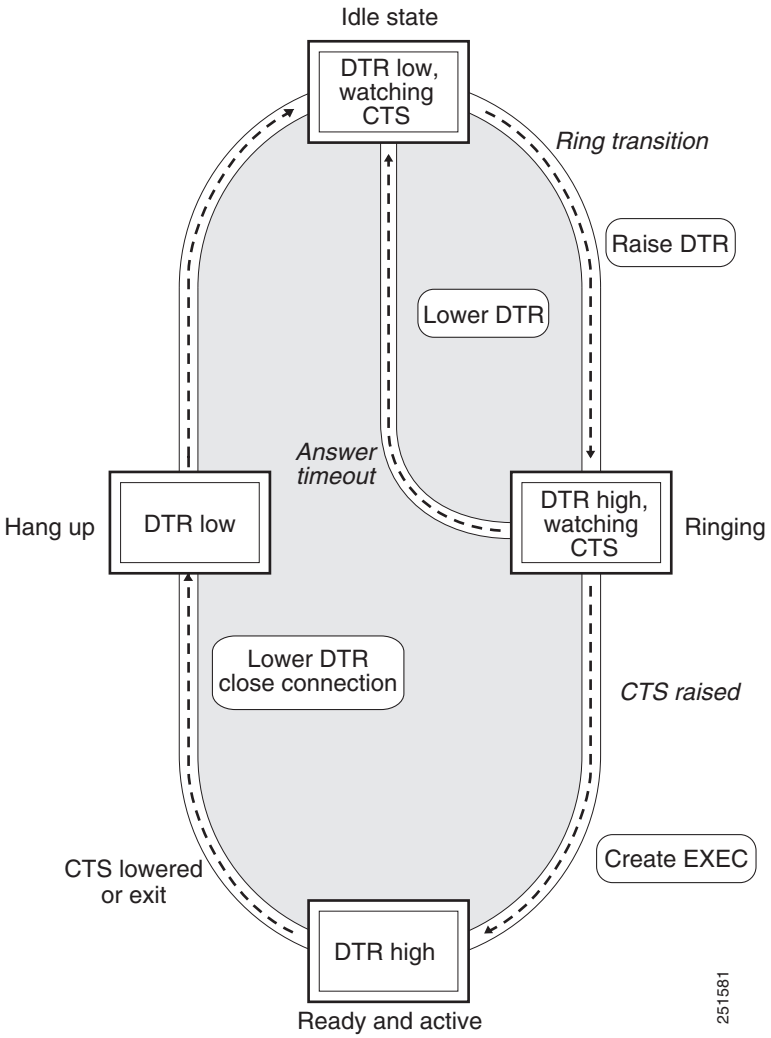
Figure 1 EXEC and Daemon Creation on a Line with No Modem Control



EXEC Creation on a Line Configured for a High-Speed Modem

Figure 2 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.

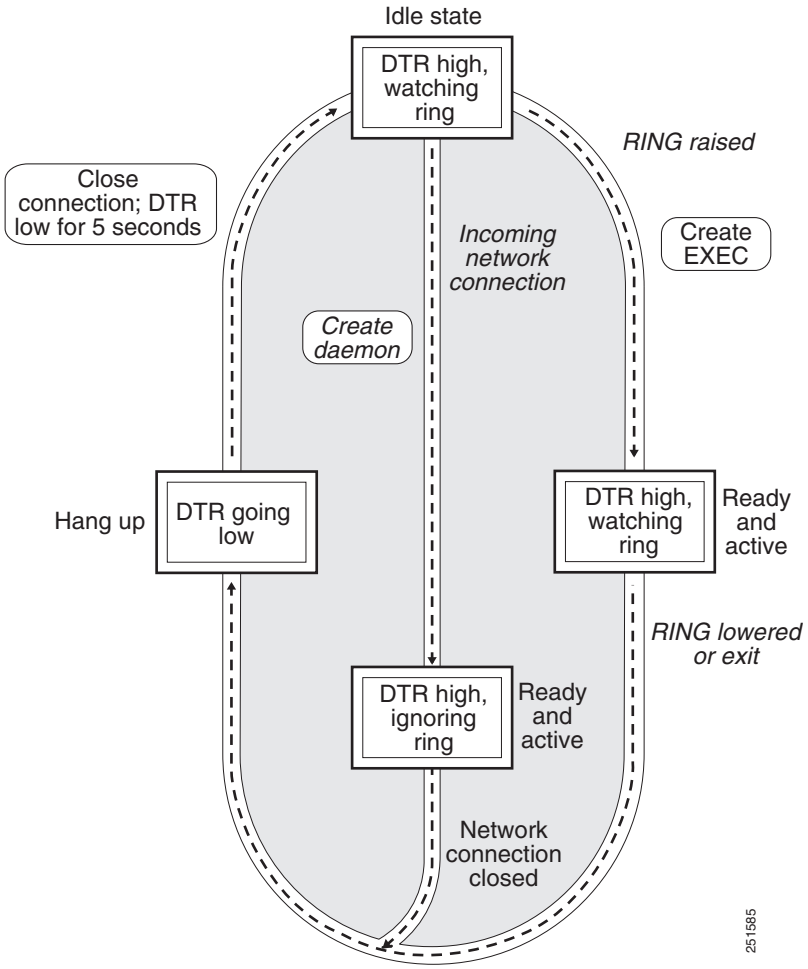
Figure 2 EXEC Creation on a Line Configured for a High-Speed Modem



EXEC and Daemon Creation on a Line for Incoming and Outgoing Calls

Figure 3 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". If the line is activated by an incoming TCP connection, the line functions similarly to lines not used with modems.

Figure 3 EXEC and Daemon Creation on a Line for Incoming and Outgoing Calls

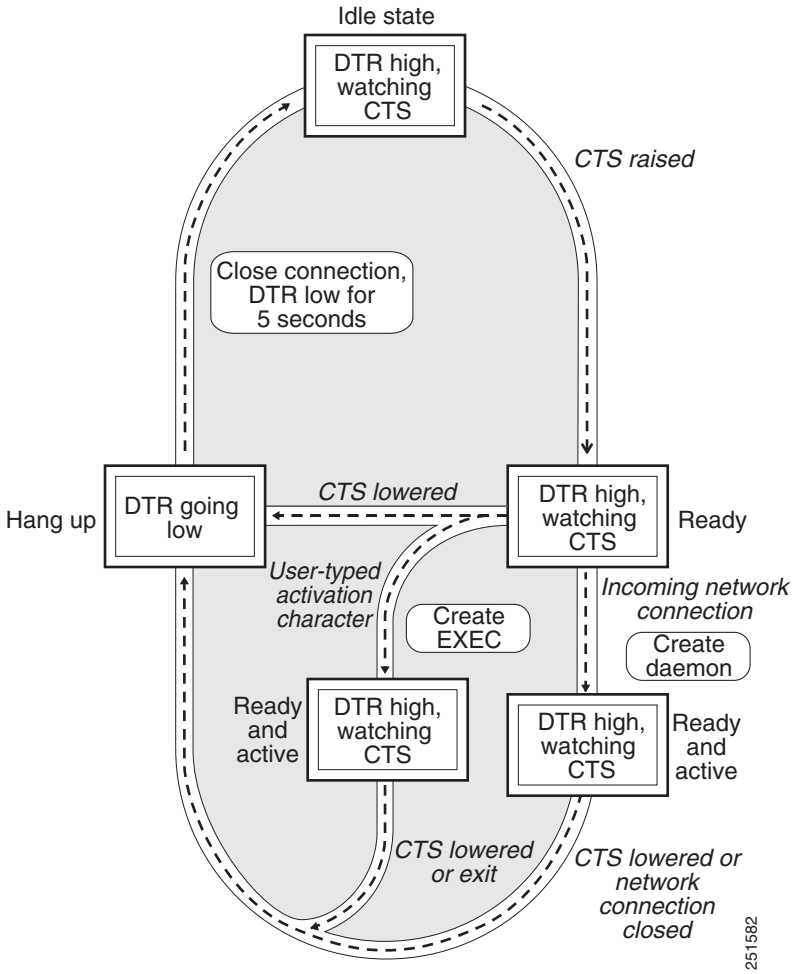


Note If your system incorporates dial-out modems, consider using access lists to prevent unauthorized use.

EXEC and Daemon Creation on a Line Configured for Continuos CTS

Figure 4 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).

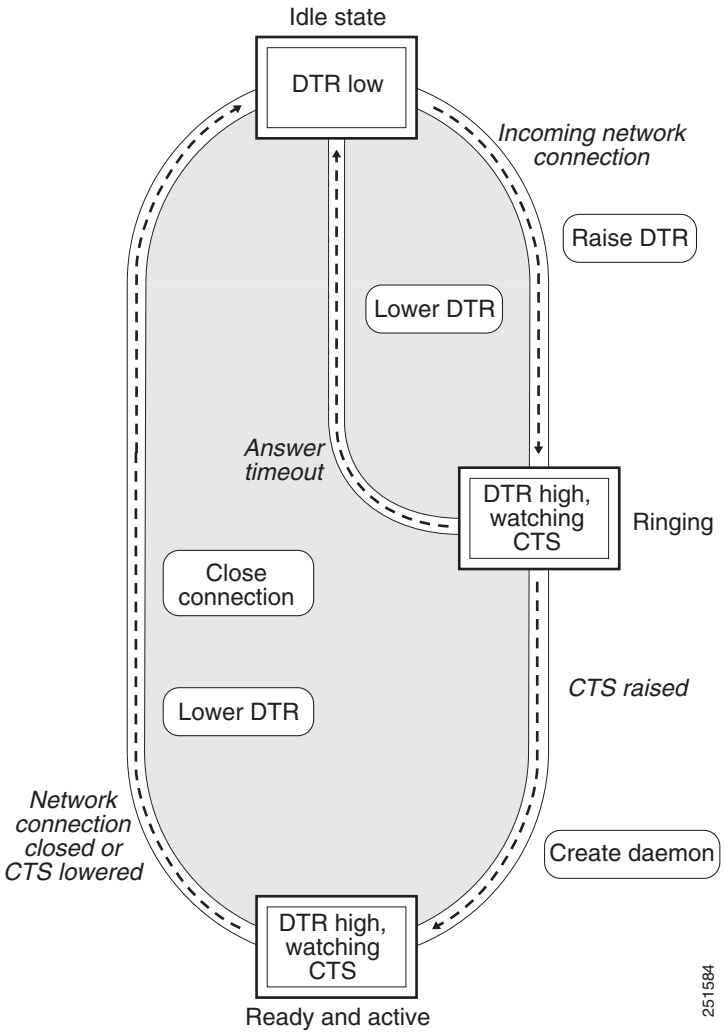
Figure 4 EXEC and Daemon Creation on a Line Configured for Continuous CTS



Daemon Creation on a Line Configured for Modem Dial-Out

Figure 5 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 5 Daemon Creation on a Line Configured for Modem Dial-Out



How to Configure Modem Signal and Line States

To configure modem signal and line states, complete the tasks in the following sections:

Supporting EXEC Restarts Triggered Via the Clear to Send (CTS) Hardware Line State

The **modem cts-alarm** command enables the router to react to a CTS drop from the remote device, and to clear any existing EXEC session.

The router reacts to a CTS drop from a connected asynchronous device. When a CTS drop is detected, the existing EXEC session is cleared and there is no need to wait for a timeout. This method improves the speed EXEC recovery by using hardware signals.

Note Use this feature with an asynchronous serial device that relies only on CTS for flow control. The CTS performs a role similar to that of on-hook and off-hook functionality.

To enable the router to react to a Clear to Send (CTS) drop from a remote device, and to clear an existing EXEC session, use the **modem cts-alarm** command in line configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **modem cts-alarm**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)#` **line** *line-number*	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)#` **modem cts-alarm**	Configures the router to react on a CTS drop from the remote device, and to clear an existing EXEC session.

Configuring Automatic Dialing

The **modem dtr-active** command enables the router to initiate automatic dialin.

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.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **modem dtr-active**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# line` *line-number*	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)# modem dtr-active`	Configures a line to initiate automatic dialing.

Automatically Answering a Modem

The **modem dialin** command allows the router to 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 **modem dialin** command in line configuration mode.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **modem dialin**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# line` *line-number*	Enters line configuration mode for the line number specified.
Step 4	`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.

Supporting Dial-In and Dial-Out Connections

The **modem inout** command enables the router to configure a line for both incoming and outgoing calls.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **modem inout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# line line-number`	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)# modem inout`	Configures a line for both incoming and outgoing calls.

Configuring a Line Timeout Interval

The **modem answer-timeout** command enables the router 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).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **modem answer-timeout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# line` *line-number*	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)# modem answer-timeout`	Configures a line for both incoming and outgoing calls.

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.

The **modem cts-required** enables the router 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **modem answer-timeout**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)#` **line** *line-number*	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)#` **modem cts-required**	Configures a line to close connections.

Configuring a Line to Disconnect Automatically

The **autohangup** command enables the router to configure 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **autohangup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# line` *line-number*	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)# autohangup`	Configures automatic line disconnect.

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. The **modem callout** command enables the router to configure the Cisco IOS software to function somewhat like a modem, and prevents the incoming calls.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** *line-number*
4. **autohangup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	`Router(config)# line line-number`	Enters line configuration mode for the line number specified.
Step 4	`Router(config-line)# modem callout`	Configures a line for reverse connections and prevents incoming calls.

Additional References

The following sections provide references related to the Modem Signal and Line State feature.

Related Documents

Related Topic	Document Title
Modem Configuration Commands	*Cisco IOS Dial Technologies Command Reference*
Modem Configuration and Management	*Cisco IOS Dial Technologies Configuration Guide*

Standards

Standard	Title
None	

MIBs

MIB	MIBs Link
•	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Dial Command Reference* at http://cisco.com/en/US/docs/ios/dial/command/reference/dia_book.html. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or the *Cisco IOS Master Command List, All Releases*, at http://www.cisco.com/en/US/docs/ios/mcl/allreleasemcl/all_book.html.

- **modem cts-alarm**
- **modem dtr-active**
- **modem dialin**
- **modem inout**
- **modem answer-timeout**
- **modem cts-required**
- **modem callout**

Feature Information for Modem Signal and Line State

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Modem Signal and Line State

Feature Name	Releases	Feature Information
Automatic Modem Configuration	11.2(1) 12.0(2)T 12.0(7)T 12.2(11)YT 12.2(11)YV 12.2(13)T 12.2(4)T 12.2(8)T 12.5 12.4T 12.2SX	Automatic Modem Configuration can issue initialization strings automatically for most types of modems externally attached to the access server.

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

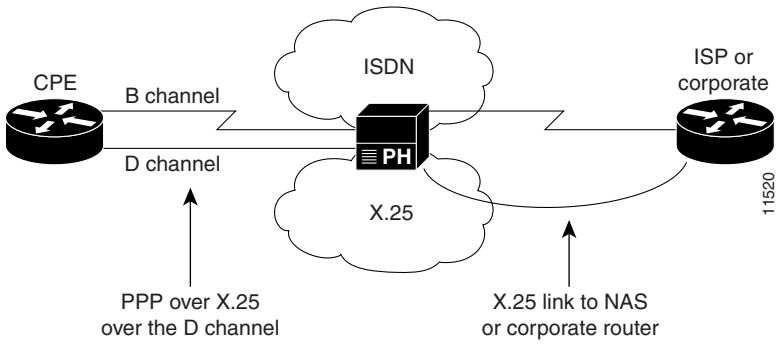
- BACP/BAP

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 1 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.

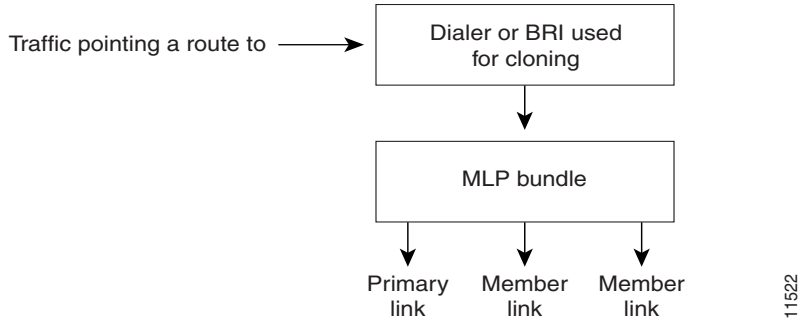
Figure 1 AO/DI Environment



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 2.)

Figure 2 Increasing Port Efficiency with AO/DI

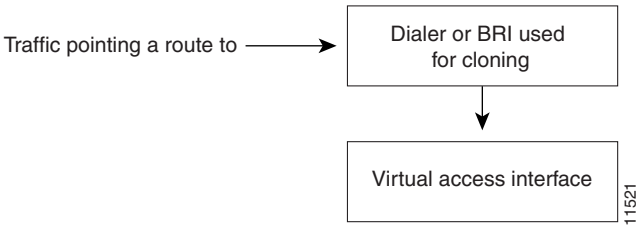


PPP over X.25 Encapsulation

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 3 shows the virtual interface creation process.

Figure 3 Creating a Virtual Access Interface



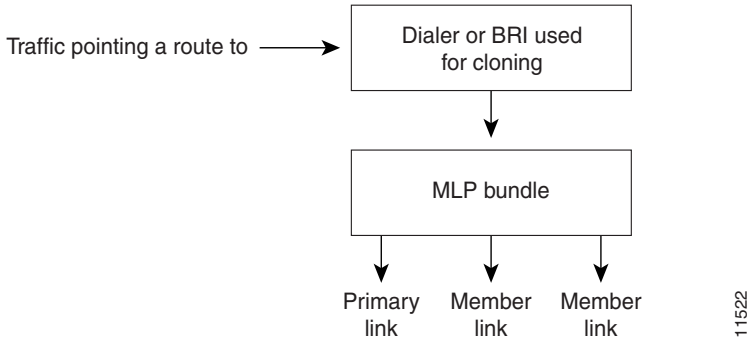
Multilink PPP Bundle

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 4 shows the MLP bundle sequence.

Figure 4 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.
`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.

Command	Purpose
`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.
`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.

Command	Purpose
`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
```

Configuring ISDN BRI

First Published: February 26, 2003
Last Updated: November 24, 2010

This module describes tasks that are required to use an Integrated Services Digital Network (ISDN) Basic Rate Interface (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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Configuring ISDN BRI" section on page 41.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Information About ISDN BRI

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. 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 kb/s, and one 16 kb/s 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 kb/s, but many users use only half the available bandwidth. This section covers the following topics:

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).
- Ask for a point-to-point service and a data-only line if the router will be the only device attached to the BRI.
- Ask for point-to-multipoint service (subaddressing is required) and a voice-and-data line if the router will be attached to an ISDN bus (to which other ISDN devices might be attached).

When you order ISDN service for switches used in North America, request the BRI switch configuration attributes specified in Table 1.

Table 1 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 kb/s outside local exchange. Directory number 1 can hunt to directory number 2.

Table 1 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 kb/s 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 kb/s 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 kb/s 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. The interface provides capabilities like call screening, called party-number verification, and ISDN default cause code override. For European and Australian customers, the interface provides Dialed Number Identification Service (DNIS)-plus-ISDN-subaddress binding to allow multiple binds between a dialer profile and an ISDN B channel.

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.

This section covers the following topics:

Dynamic Multiple Encapsulations

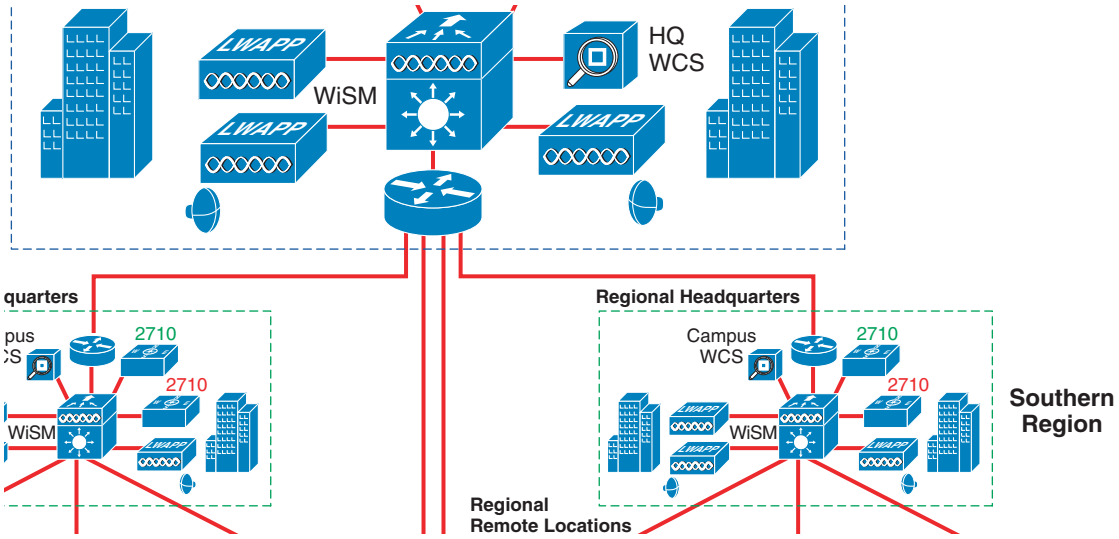
Prior to Cisco IOS Release 12.1, encapsulation techniques such as Frame Relay, High-Level Data Link Control (HDLC), Link Access Procedure, Balanced-Terminal Adapter (LAPB-TA), and X.25 could support only one ISDN B-channel connection over the entire link. HDLC and PPP could support multiple B channels, 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 depending on 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 the maximum use of all the 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 1 shows a typical configuration for an X.25 network in Europe. The Dynamic Multiple Encapsulations feature allows the use of all the 30 B channels, and supports calls that originate in diverse areas of the network and converge on the same ISDN PRI.

Figure 1 European X.25 Network



Interface Configuration Options

You can also optionally configure the snapshot routing for the ISDN interfaces. Snapshot routing is a method of dynamically learning remote routes and keeping the routes available for a specified period of time, even though routing updates are not exchanged during that period.

To place calls on an ISDN interface, you must configure the interface with dial-on-demand routing (DDR). For configuration information about ISDN by 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 the bandwidth on demand, see the modules "Configuring Legacy DDR Spokes" or "Configuring Legacy DDR Hubs" in the *Cisco IOS Dial Solutions Configuration Guide*.

ISDN Cause Codes

A cause code is an information element (IE) that indicates why an ISDN call failed or was disconnected. When the originating gateway receives a Release Complete message, it generates a tone corresponding to the cause code in the message.

Table 2 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 2 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

Table 2 Cause Codes Generated by the Cisco VoIP Gateway (continued)

Cause Code	Description	Explanation	Tone
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 2 by default, there are commands introduced in previous Cisco IOS releases that can override these defaults, thereby 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 following tasks:

- Configuring the ISDN BRI Switch, page 7 (required)
- Specifying Interface Characteristics for an ISDN BRI, page 10 (required)
- Configuring ISDN Semipermanent Connections, page 29 (required)
- Configuring ISDN BRI for Leased-Line Service, page 31 (required)
- Monitoring and Maintaining ISDN Interfaces, page 33 (optional)
- Troubleshooting ISDN Interfaces, page 34 (optional)

See the Monitoring and Maintaining ISDN Interfaces sections on page 452 and the "Troubleshooting ISDN Interfaces" section on page 453 in this module for tips on maintaining your network. See the section "Configuration Examples for Configuring ISDN BRI" section on page 455 for sample configurations.

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, page 7 (required)
- Checking and Setting the Buffers, page 8 (required)
- Configuring Buffers and MTU Size, page 9 (required)

Configuring the Switch Type

Perform this task to configure the switch type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isdn switch-type** *switch-type*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**isdn switch-type** *switch-type* **Example:** `Router(config)# isdn switch-type basic-ni`	Selects the service provider switch type. See Table 3 for valid switch type keywords.
Step 4	**end** **Example:** `Router(config# end`	Exits global configuration mode.

The "Example: Configuring a Global ISDN and BRI Interface Switch Type" section on page 454 provides an example of configuring the ISDN BRI switch.

Table 3 lists the ISDN BRI service provider switch type keywords.

Table 3 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** and **vn3** switch type keywords are displayed respectively.

Checking and Setting the Buffers

When configuring a BRI interface, after the system starts up, make sure that the free list of the buffer pool has enough buffers that match the maximum transmission unit (MTU) of your BRI interface. If not, you must reconfigure buffers in order for the BRI interfaces to function properly.

Perform this task to check the MTU size and buffers.

SUMMARY STEPS

1. **enable**
2. **show interfaces bri** *number*
3. **show buffers**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router# enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interfaces bri** *number* **Example:** `Router# show interfaces bri 0`	Displays the MTU size.
Step 3	**show buffers** **Example:** `Router# show buffers`	Displays the free buffers.
Step 4	**end** **Example:** `Router# end`	Exits privileged EXEC configuration mode.

Configuring Buffers and MTU Size

Perform this task to configure the buffers and the MTU size.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **buffers** {{**header** | **fastswitching** | *interface number* | **small** | **middle** | **big** | **verybig** | **large** | **huge** {**initial** | **max-free** | **min-free** | **permanent**} **buffers**} | **particle-clone** *particle-clones* | **element** {**minimum** | **permanent**} **elements**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**buffers** {{**header** | **fastswitching** | *interface number* | **small** | **middle** | **big** | **verybig** | **large** | **huge** {**initial** | **max-free** | **min-free** | **permanent**} **buffers**} | **particle-clone** *particle-clones* | **element** {**minimum** | **permanent**} **elements**} **Example:** `Router(config)# buffers big`	Configures the size of the buffer and the initial public buffer pool settings.
Step 4	**end** **Example:** `Router# end`	Exits global configuration mode.

Specifying Interface Characteristics for an ISDN BRI

Perform the following tasks to set interface characteristics for an ISDN BRI interface irrespective of 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

Perform this task to enter interface configuration mode and specify an ISDN BRI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** bri *number*
4. **ip address** *address mask*
5. **end**

DETAILED STEPS

	Command	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

Step 3	**interface bri** *number*
	
	Example:
	`Router(config)# interface bri 0`
	
	Cisco 7200 series router only
	`Router(config)# interface bri slot/port`
		Specifies the interface and enters interface configuration mode.

Step 4	**ip address** *address mask* **Example:** `Router(config-if)# ip address ip 209.165.200.225 255.255.255.224`	Specifies an IP address for the interface.
Step 5	**end** **Example:** `Router# end`	Exits interface configuration mode.

Specifying ISDN SPIDs

Some service providers use service profile identifiers (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 the ISDN service without SPIDs. In addition, SPIDs have significance only at the local access ISDN interface. 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. The DMS-100 switch type has two SPIDs — one for each B channel.

The **isdn spid1** and **isdn spid2** commands enable the router to define the SPIDs and the local directory number (LDN) on the router.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface bri** *number*
4. **isdn spid1** *word*
5. **isdn spid2** *word*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.
Step 4	**isdn spid1** *word* **Example:** `Router(config-if)# isdn spid1 415988488201`	Specifies a SPID and a name for the B1 channel.
Step 5	**isdn spid2** *word* **Example:** `Router(config-if)# isdn spid2 415988488302`	Specifies a SPID and a name for the B2 channel.
Step 6	**end** **Example:** `Router# end`	Exits interface configuration mode and returns to global configuration mode.

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.

Perform this task to configure encapsulation on the ISDN BRI.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **encapsulation** [**ppp** | **lapb** | **hdlc** | **x25**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**encapsulation** [**ppp** | **lapb** | **hdlc** | **x25**] **Example:** `Router(config-if)# encapsulation ppp`	Configures the encapsulation type.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Verifying the Dynamic Multiple Encapsulations Feature

To verify dialer interfaces configured for binding and to see statistics on each physical interface bound to the dialer interface, use the **show interfaces** command.

The following example shows that the output under the B channel keeps all the 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 :1

: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 interface 0.

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.

To configure the router for automatic detection of encapsulation type on incoming calls, or to configure encapsulation for Cisco 700 and 800 series (formerly Combined) router compatibility, see the section "Configuring Automatic Detection of Encapsulation Type" in the module "Configuring ISDN Special Signaling" in this publication.

Configuring Network Addressing

Perform this task to configure network addressing.

This task supports the primary goals of network addressing:

- Define the packets that are interesting and those that will 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 be used 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **dialer map** *protocol-keyword protocol-next-hop-address* **name** *hostname* **speed** [**56** | **64**] *dial-string*[**:***isdn-subaddress*]

   or

   dialer map *protocol next-hop-address* **name** *hostname* **spc** [**speed 56** | **64**] [**broadcast**] *dial-string*[**:***isdn-subaddress*]
5. **dialer-group** *group-number*

6. **exit**
7. **dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-numbe*r | *access-group*}
8. **access-list** *access-list-number* {**deny** | **permit**} *protocol source address source-mask destination destination-mask*
9. end

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.

Step 4	**dialer map** *protocol-keyword protocol-next-hop-address* **name** *hostname* **speed** [**56** | **64**] *dial-string*[**:***isdn-subaddress*]
	
	Example:
	`Router(config-if)# dialer map 172.19.1.8 name abc speed 64 zzz 14155550134`
	
	or
	
	dialer map *protocol next-hop-address* **name** *hostname* **spc** [**speed 56** | **64**] [**broadcast**] *dial-string*[**:***isdn-subaddress*]
	
	Example:
	`Router(config-if)# dialer map ip 172.16.0.0 name user1 spc 64 broadcast zzz 14155550134`
	
	or
	
	Example:
	`Router (config-if)# dialer map ip 172.16.0.0 name user2 spc 64 broadcast yyyy 14155550134`
		(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 5	**dialer-group** *group-number* **Example:** `Router(config-if)# dialer-group 1`	Assigns the interface to a dialer group to control access to the interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits to global configuration mode.

	Command or Action	Purpose
Step 7	**dialer-list** *dialer-group* **protocol** *protocol-name* {**permit** | **deny** | **list** *access-list-number* |*access-group*} **Example:** `Router(config)# dialer-list 1 protocol ip list 101`	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 8	**access-list** *access-list-number* {**deny** | **permit**} *protocol source address source-mask destination destination-mask* **Example:** `Router(config)# access-list 202 deny ip 192.0.2.0 255.255.255.224 192.0.2.3 255.255.255.224`	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 9	**exit** **Example:** `Router(config)# exit`	Exits global configuration mode.

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 8 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 module 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 modules "Configuring Legacy DDR Hubs" and "Configuring Legacy DDR Spokes" in the *Cisco IOS Dial Solutions Configuration Guide*.

Configuring TEI Negotiation Timing

Perform this task to configure terminal endpoint identifier (TEI) negotiation timing.

The **isdn tei** command enables the router to apply the TEI negotiation to a specific BRI interface.

You can configure the ISDN TEI negotiation on individual ISDN interfaces. The 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 the 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 the 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**).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn tei** [**first-call** | **powerup** | **preserve** | **remove**]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.
Step 4	**isdn tei** [**first-call** | **powerup** | **preserve** | **remove**] **Example:** `Router(config-if)# isdn tei first-call`	Determines when ISDN TEI negotiation occurs.
Step 5	**end** **Example:** `Router(config-if)# exit`	Exits interface configuration mode and returns to global configuration mode.

Configuring CLI Screening

Perform this task to configure 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn caller** *word*

5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** number Example: `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**isdn caller** *word* **Example:** `Router(config-if)# isdn caller 415988488201`	Configures caller ID screening.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Note If caller ID screening is configured and the local switch does not deliver caller IDs, the router rejects all calls.

Note In releases prior to Cisco IOS Release 12.1, ISDN accepted all synchronous calls and performed some minimal CLI screening before accepting or rejecting a call. Beginning with Cisco IOS Release 12.1, 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

Perform this task to configure 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 verifies the 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn answer1** [*called-party-number*] [**:***subaddress*]
5. **isdn answer2** [*called-party-number*] [**:***subaddress*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.
Step 4	**isdn answer1** [*called-party-number*] [**:***subaddress*] **Example:** `Router(config-if)# isdn answer1 [123][:56789]`	Specifies that the router verify a called-party number or subaddress number in the incoming setup message.
Step 5	**isdn answer2** [*called-party-number*] [**:***subaddress*] **Example:** `Router(config-if)# isdn answer2 [567][:45903]`	Specifies that the router verify a second called-party number or subaddress number in the incoming setup message.
Step 6	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

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.

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 that display the number. When configured, this information is included in the outgoing call Setup message.

Perform this task to configure the interface to identify the billing number.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn calling-number** *word*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**isdn calling-number** *word* **Example:** `Router(config-if)# isdn calling-number 415988488201`	Specifies the calling party number.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

The **isdn calling-number** 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

Perform this task to configure the line speed for calls that are not ISDN from end to end.

When calls are made at 56 kb/s but delivered by the ISDN network at 64 kb/s, the incoming data can get corrupted. However, on ISDN calls, if the receiving side is informed that the call is not an ISDN call from end to end, the ISDN network can set the line speed for the incoming call.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn not-end-to-end** {**56** | **64**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**isdn not-end-to-end** {**56** | **64**} **Example:** `Router(config-if)# isdn not-end-to-end 56`	Sets the speed to be used for incoming calls recognized as not ISDN end to end.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Configuring a Fast Rollover Delay

Perform this task to configure a fast rollover delay.

Sometimes a router attempts to dial a call on an ISDN B channel before the 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 also need to configure this delay if you have a dialer map for each phone number and the first call succeeds but a second call fails with no channel available.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn fast-rollover-delay** *seconds*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.
Step 4	**isdn fast-rollover-delay** *seconds* **Example:** `Router(config-if)# isdn fast-rollover-delay 56`	Defines a fast rollover delay.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

A delay of 5 seconds should cover most cases. Configure sufficient delay to ensure 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

Perform this task to override 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.

SUMMARY STEPS

1. **enable**

2. **configure terminal**
3. **interface bri** *number*
4. **isdn disconnect-cause** {*cause-code-number* | **busy** | **not-available**}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.
Step 4	**isdn disconnect-cause** {*cause-code-number* | **busy** | **not-available**} **Example:** `Router(config-if)# isdn disconnect-cause not-available`	Specifies the ISDN cause code to be sent to the switch.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Examples

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

Perform this task to configure 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn sending-complete**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**isdn sending-complete** **Example:** `Router(config-if)# isdn sending-complete`	Includes the Sending Complete information element in the outgoing call Setup message.
Step 5	**end** **Example:** `Router(config-if)# end`	Exits interface configuration mode and returns to global configuration mode.

Configuring DNIS-plus-ISDN-Subaddress Binding

Perform this task to configure DNIS-plus-ISDN-subaddress binding.

SUMMARY STEPS

1. **enable**
2. **configure terminal**

3. **dialer called** *dnis*:*subaddress*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**dialer called** *dnis*:*subaddress* **Example:** `Router(config)# dialer called 12345:6789`	Binds a DNIS to an ISDN subaddress.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Note The **dialer called** 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 "Example: DNIS-plus-ISDN-Subaddress Binding" on page 458 in this module for a configuration example.

Screening Incoming V.110 Modem Calls

Perform this task to screen incoming V.110 modem calls. You can screen incoming V.110 modem calls and reject calls that do not have the communications settings configured according to the network.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **isdn v110 only** [**databits** {**5** | **7** | **8**}] [**parity** {**even** | **mark** | **none** | **odd** | **space**}] [**stopbits** {**1** | **1.5** | **2**}]
5. **end**

DETAIL ED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**isdn v110 only** [**databits** {**5** | **7** | **8**}] [**parity** {**even** | **mark** | **none** | **odd** | **space**}] [**stopbits** {**1** | **1.5** | **2**}] **Example:** `Router(config-if)# isdn v110 only databits 8 parity none stopbits 1`	Selectively accepts incoming V.110 calls based on data bit, parity, and stop bit modem communication settings.
Step 5	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Disabling V.110 Padding

Perform this task to disable 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **no isdn v110 padding**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and enters interface configuration mode.
Step 4	**no isdn v110 padding** **Example:** `Router(config-if)# no isdn v110 padding`	Disables the padded modem speed report required by the V.110 modem standard.
Step 5	**end** **Example:** `Router(config)# end`	Exits interface configuration mode and returns to global configuration mode.

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 better priced than leased lines.

Configuring BRI interfaces for semipermanent connection requires only a keyword that indicates the semipermanent connections when you are setting up network addressing as described in the previous section of this module.

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". For ISDN PRI, see the section "How to Configure ISDN PRI" in the module "Configuring ISDN PRI".

Step 2 Configure DDR on a selected interface, as described in the "Dial-on-Demand Routing Configuration" part of this publication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*

4. **dialer map** *protocol next-hop-address* **name** *hostname* **spc** [**speed 56** | **64**] [**broadcast**] *dial-string*[**:***isdn-subaddress*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface bri** *number* **Example:** `Router(config)# interface BRI 0`	Specifies the interface and begins interface configuration mode.
Step 4	**dialer map** *protocol next-hop-address* **name** *hostname* **spc [speed 56 | 64]** [**broadcast**] *dial-string*[**:***isdn-subaddress*] **Example:** `Router(config-if)# dialer map ip 172.19.1.8 name user1 spc 64 [broadcast] zzz 14155550134`	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 kb/s, as needed.
Step 5	**end** **Example:** `Router(config)# end`	Exits interface configuration mode and returns to global configuration mode.

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 Kb/s (Available only in Japan)

Note When 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.

The **isdn leased-line bri** command enable the router to configure the BRI to use the ISDN connection as a leased-line service. The **no isdn leased-line bri command** removes the leased-line configuration from a specified ISDN BRI interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isdn leased-line bri** *number number* [**128** | **144**]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**isdn leased-line bri** *number number* [**128** | **144**] **Example:** `Router(config)# isdn leased-line BRI 0 123 [128 | 144]`	Specifies the combined B1 and B2 channels leased lines.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Configuring Leased-Line Service at 128 Kb/s

The Cisco IOS software supports leased-line service at 128 kb/s via ISDN BRI. 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 kb/s. When this software first became available, service providers offered support for ISDN channel aggregation at 128 kb/s only in Japan.

The **isdn leased-line bri** commands enable the router to configure the BRI to use the ISDN connection as a leased-line service at 128kb/s.

Note This feature is not supported on the Cisco 2500 series router because its BRI hardware does not support channel aggregation.

The **no isdn leased-line bri command** command removes the leased-line configuration from a specified ISDN BRI interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isdn leased-line bri** *number* **128**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**isdn leased-line bri** *number* **128** **Example:** `Router(config)# isdn leased-line BRI 0 128`	Configures a specified BRI for access over leased lines.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

To complete the configuration of the interface, see the module "Configure Synchronous Serial Ports" in this module.

Monitoring and Maintaining ISDN Interfaces

The **show interfaces**, **show controllers**, **show isdn,** and the **show dialer interface bri** commands enable the router to monitor and maintain ISDN interfaces, use the following commands in EXEC mode as needed:

SUMMARY STEPS

1. **enable**
2. **show interfaces bri** *number*
3. **show controllers bri** *number*
4. **show isdn** {**active** | **history** | **memory** | **status** | **timers**}
5. **show dialer interface bri** *number*
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**show interfaces** ***bri*** *number*
	
	Cisco 7200 series routers only
	show interfaces bri slot/port
	
	Example:
	`show interfaces bri 0`
	
	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.

Step 3	**show controllers bri** *number* **Cisco 7200 series routers only** **show controllers bri** *slot/port* **Example:** `Router> show controllers bri 0` **Cisco 7200 series routers only** `Router> show controllers bri 0`	Displays protocol information about the ISDN B and D channels.
Step 4	**show isdn** {**active** | **history** | **memory** | **status** | **timers**} **Example:** `Router> show isdn active`	Displays information about calls, history, memory, status, and Layer 2 and Layer 3 timers.
Step 5	**show dialer interface bri** *number* **Example:** `Router> show dialer interface bri 0`	Obtains general diagnostic information about the specified interface.
Step 6	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Troubleshooting ISDN Interfaces

The following commands can help verify the ISDN configuration of the router:

- **show controllers bri** *number*—Checks Layer 1 (physical layer) of the BRI.
- **debug q921**—Checks Layer 2 (data link layer).
- **debug isdn events**—Checks the network layer.
- **debug q931**—Checks the network layer.

- **debug dialer**—Checks the network layer.
- **show dialer**—Checks the network layer.

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

Configuration Examples for Configuring ISDN BRI

This section provides the following ISDN BRI configuration examples:

Example: Configuring a Global ISDN and BRI Interface Switch Type

The following example shows to configure 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 bri 0
 isdn switch-type basic-net3
```

Example: Configuring a BRI Connected to a PBX

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 bri 0
 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
```

Example: Configuring Multilink PPP on a BRI Interface

The following example shows how to enable Multilink PPP on BRI interface 0:

```
interface bri 0
 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
```

Example: Configuring Dialer Rotary Groups

The following example shows how to configure BRI interfaces to connect into a rotary group (using the **dialer-group** command). It also shows how to configure 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
```

Example: Configuring Predictor Compression

The following example shows how to enable predictor compression on BRI interface 0:

```
interface bri 0
 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 shows how to enable stacker compression on BRI interface 0:

```
interface bri 0
 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
```

Example: Configuring Multilink PPP and Compression

The following example shows how to enable Multilink PPP and stacker compression on BRI interface 0:

```
interface bri 0
 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
```

Example: Configuring Voice over ISDN

The following example shows how to allow incoming voice calls to be answered on BRI interface 0:

```
interface bri 0
 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 shows how to allow outgoing voice calls on BRI interface 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*.

Example: DNIS-plus-ISDN-Subaddress Binding

The following example shows how to configure 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 module "Configuring Peer-to-Peer DDR with Dialer Profiles" of this publication.

Example: Screening Incoming V.110 Modem Calls

The following example shows to filter 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
```

Example: ISDN BRI Leased-Line Configuration

The following example shows how to configure BRI interface 0 for leased-line access at 128 kb/s. Because of the leased-line—not dialed—environment configuration of ISDN called and calling numbers are not needed and not used. BRI interface 0 is henceforth treated as a synchronous serial interface, with the default HDLC encapsulation.

```
isdn leased-line 128
```

Additional References

Related Documents

Related Topic	Document Title
Modem configuration commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Dial Technologies Command Reference*
Modem configuration and management	*Cisco IOS Dial Technologies Configuration Guide*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
No new or modified MIBs are supported, and support for existing MIBs has not been modified.	To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
No new or modified RFCs are supported, and support for existing RFCs has not been modified.	

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring ISDN BRI

Table 4 lists the release history for this feature.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 4 lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 4 Feature Information for Modem Signal and Line State

Feature Name	Releases	Feature Information
National ISDN Switch Types for BRI and PRI Interfaces	11.3(3)T 12.0(1) 12.0(1)T 12.1(14) 12.1(3)T 12.2(11)YT 12.2(11)YV 12.2(13)T 12.2(15)T 12.2(2)T 12.2(8)T 12.5 12.4T 12.2SR 15.1.(1)S	The National ISDN Switch Types for Basic Rate and Primary Rate Interfaces feature introduces changes to ISDN switch types for Primary Rate Interfaces (PRI) and BRI. These switches provide the ability to connect to multiple ISDN switch types (BRI and PRI) and the NI2 switch type. The following commands were introduced or modified: **dialer called**, **dialer map**, **isdn calling-number**, **isdn disconnect-cause**, **isdn fast-rollover-delay**, **isdn leased-line type**, **isdn not-end-to-end**, **isdn sending complete**, **isdn v10 only**, **no isdn v10 padding**, **show controllers show isdn**, **show dialer interface bri**, **show interfaces**.

Leased and Switched BRI Interfaces for ETSI NET3

Feature History

Release	Modification
12.2(4)T	This feature was introduced on the Cisco 800 series routers.

This document describes the Leased and Switched BRI Interfaces for ETSI NET3 feature. It includes the following sections:

Feature Overview

In most BRI configurations, both B channels of a leased-line service are used as point-to-point leased lines with the D channel disabled. Data transmission over the B channels is no different than data transmission over point-to-point leased lines.

A new feature available in Cisco IOS Release 12.2(4)T, Leased and Switched BRI Interfaces for ETSI NET3, allows one BRI B channel on a European Telecommunications Standards Institute (ETSI) NET3 switch to be configured as a leased line, and the second B channel to be configured as a standard ISDN or dial interface and used as a switched channel to the Public Switched Telephone Network (PSTN). When the Leased and Switched BRI Interfaces for ETSI NET3 feature is configured, one B channel functions as a point-to-point 64-kbps leased line and the other B channel functions as a circuit-switched channel using the D channel to provide the signaling features available for the ETSI NET3 signaling protocol.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits

The Leased and Switched BRI Interfaces for ETSI NET3 feature allows Internet service providers to split one ISDN line into a leased-line interface and a dialer interface, thereby increasing connection capability without increasing cost.

Restrictions

The following restrictions apply to the Leased and Switched BRI Interfaces for ETSI NET3 feature:

- Only the ETSI NET3 signaling protocol is supported at a line speed of 64 kbps.
- Only one ISDN call can be active at any time, and the call must verify that the leased line is not used to bring up a second call.
- The ETSI NET3 switch cannot be configured for a leased line when the U interface is used instead of the S/T interface; doing so prevents the line protocol from coming up.

Related Documents

- *Cisco IOS Dial Technologies Command Reference*, Release 12.2
- *Cisco IOS Dial Technologies Configuration Guide*, "ISDN Configuration" part, Release 12.2

Supported Platforms

- Cisco 800 series

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

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

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

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

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

The fillin_isdnBearerEntry() – isdnBearerTable manipulation MIB function is supported. See the "RFCs" section for more information.

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 the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

- RFC 2127, *ISDN Management Information Base using SMIv2*

  RFC 2127 states that the following be reported for the leased line B channel: The isdnBearerTable MIB entry for the leased line B channel will need to be altered, specifically the value of isdnBearerChannelType MIB will be set to leased(2). This alteration involves a function in isdn/sr_ietf_isdmib.c, namely the fillin_isdnBearerEntry() — isdnBearerTable manipulation MIB function.

- RFC 1573, *Evolution of the Interfaces Group of MIB-II*

  RFC 1573 makes no explicit mention of changes to the ifEntry for a B channel set to leased line. It is proposed that the ifAdminStatus and ifOperStatus functions remain in the UP(1) state.

Prerequisites

Before starting the configuration tasks in this document, review the chapter "Configuring ISDN BRI" and the section "Configuring ISDN BRI for Leased-Line Service," for more complete details on configuring a BRI. This chapter is in the part "ISDN Configuration" in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the Leased and Switched BRI Interfaces for ETSI NET3 feature. Each task in the list is identified as either required or optional:

- Configuring Leased and Switched BRI Interfaces for ETSI NET3 (required)
- Verifying Leased and Switched BRI Interfaces for ETSI NET3 (optional)

Configuring Leased and Switched BRI Interfaces for ETSI NET3

To configure a BRI for both an ISDN connection and leased-line service, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# isdn switch-type basic-net3`	Configures the ETSI NET3 BRI switch type.
Step 2	`Router(config)# isdn leased-line bri`*number*`/`*number* {**b1** | **b2**}	Splits a line for both ISDN and 64-kbps leased-line service.

Verifying Leased and Switched BRI Interfaces for ETSI NET3

Note In the following verification procedure, BRI channel B1(the BRI0:1 interface) is configured for leased-line service and channel B2 (the BRI0:2 interface) is configured for ISDN.

To verify that each BRI channel is configured correctly, perform the following steps:

Step 1 Enter the **show isdn status** EXEC command and check the value in The Free Channel Mask field to verify that only one channel has been allocated for ISDN. The Free Channel Mask field displays 0x80000000 when there is an active call. If no call is active, The Free Channel Mask field displays 0x80000001 and 0x80000002 for the B1 and B2 leased line configurations, respectively.

```
Router# show isdn status

Global ISDN Switchtype = basic-net3
ISDN BRI0 interface
        dsl 0, interface ISDN Switchtype = basic-net3
    Layer 1 Status:
        ACTIVE
    Layer 2 Status:
        TEI = 124, Ces = 1, SAPI = 0, State = MULTIPLE_FRAME_ESTABLISHED
        I_Queue_Len 0, UI_Queue_Len 0
    Layer 3 Status:
        1 Active Layer 3 Call(s)
        CCB:callid=8001, sapi=0, ces=1, B-chan=2, calltype=DATA
    Active dsl 0 CCBs = 1
    The Free Channel Mask:  0x80000000
    Total Allocated ISDN CCBs = 1
```

Step 2 Enter the **show dialer** EXEC command to display dialer interface statistics. Check that there is no entry for the BRI0:1 interface in the display:

```
Router# show dialer

BRI0 - dialer type = ISDN

Dial String      Successes   Failures    Last DNIS   Last status
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.

BRI0:2 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle

Di1 - dialer type = DIALER PROFILE
Idle timer (6000 secs), Fast idle timer (20 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is idle
```

```
Number of active calls = 0

Dial String      Successes   Failures    Last DNIS   Last status
5552000                  0          0    never                   -   Default
```

Troubleshooting Tips

To test the BRI configurations, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **show controllers bri** *number*	Checks Layer 1 of the BRI.
Router# **debug q921**	Checks Layer 2 of the BRI.
Router# **debug dialer**	Checks dialer events on the BRI.
Router# **debug isdn events**	Checks call control events on the BRI.
Router# **debug q931**	Checks Layer 3 of the BRI.

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

Monitoring and Maintaining Leased and Switched BRI Interfaces for ETSI NET3

To monitor and maintain the BRI configurations, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **show isdn status**	Displays Layer 1, Layer 2, and Layer 3 status of the D channel and channel mask information.
Router# **show interfaces bri** *x*:*y* **status**	Displays status of the ISDN BRI channel configured as either a leased line or dialer B-channel interface.
Router# **show dialer**	Displays status of the ISDN BRI channel configured as a dialer interface.

Configuration Examples

This section provides an example of how to configure the Leased and Switched BRI Interfaces for ETSI NET3 feature.

Leased and Switched BRI Interfaces for ETSI NET3 Example

The following example configures BRI channel B2 for 64-kbps leased-line service and channel B1 for ISDN service:

```
isdn leased-line bri0/0 b2
!
interface bri0/0
 ip address 10.1.1.1 255.255.255.0
 no ip address
 dialer pool-member 1

interface bri0/0:2
 ip address 10.1.1.2 255.255.255.0
 encapsulation ppp
 no ip address
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Dial Technologies Command Reference* at http://www.cisco.com/en/US/docs/ios/dial/command/reference/dia_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **isdn leased-line bri**

Glossary

European Telecommunications Standards Institute—See ETSI.

ETSI—European Telecommunications Standards Institute. Organization created by European Post, Telephone, and Telegraph (PTT) groups and the European Community (EC) to propose telecommunications standards for Europe.

leased line—Transmission line reserved by a communications carrier for the private use of a customer.

switched—General term applied to an electronic or mechanical device that allows a connection to be established as necessary and terminated when there is no longer a session to support.

ISDN BCAC and Round-Robin Channel Selection Enhancements

The ISDN BCAC and Round-Robin Channel Selection Enhancements feature allows more dynamic control of the ISDN B channels by providing additional B-Channel Availability Control (BCAC) functionality for configuring message signaling, and an enhanced channel selection scheme that adds round-robin configuration to the existing ascending and descending channel selection schemes already available.

Feature Specifications for the ISDN BCAC Enhancements

Feature History	
Release	**Modification**
12.3(1)	This feature was introduced.
Supported Platforms	
Cisco AS5300, Cisco AS5350, Cisco AS5400, Cisco AS5800, Cisco AS5850, Cisco 2600 series, Cisco 3640, Cisco 3660	

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for ISDN BCAC Enhancements

You need to need to be familiar with the BCAC service message signaling procedure and configuring ISDN PRI before configuring the commands described in this document. See the "Standards" section on page 10 for a list of references.

Information About the ISDN BCAC and Round-Robin Channel Selection Enhancements

The following functionality is introduced in the ISDN BCAC and Round-Robin Channel Selection Enhancements:

BCAC Enhancements

BCAC is a service message signaling procedure used to control the availability of ISDN B channels. BCAC provides a coordinated capability between both ends of a PRI to simultaneously preclude selection of specified B channels for outgoing calls, and reject calls (if channel negotiation is employed, calls may go on another channel) for those same channels. The basic BCAC functionality for the handling of SERV and SERV ACK messages already exists on Cisco routers. In Cisco IOS Release 12.3(1), the software has been enhanced with the following BCAC functionality:

- Processing of SERV and SERV ACK messages. Even though these messages are already handled in the Cisco IOS software, their processing has been enhanced to more closely align with the behavior described in the standards.
- Provides a mechanism to allow the retransmission of SERV messages.
- Handles SERV message collision cases.
- Provides service status audits for various audit triggers.
- Provides an option that when set triggers the exchange of service messages on all channels of the interface when the router is rebooted and when the signaling link comes up.
- Provides a mechanism so that if there is a flood of service messages that need to be sent, the service messages can be throttled to avoid losing them.
- Initializes B-channel service status upon provisioning.

Round-Robin Selection Scheme for ISDN B Channels

ISDN enhancements introduced in Cisco IOS Release 12.3(1) enable you to select a B channel on a PRI or a Non-Facility Associated Signaling (NFAS) interface in a round-robin fashion. This option is in addition to the ascending or descending channel selection schemes already available.

Logging of ISDN Events

ISDN enhancements introduced in Cisco IOS Release 12.3(1) support syslog logging of the following ISDN events:

- ISDN Layer 2 Up and Down events at severity 3.
- ISDN SERV, SERV ACK, RESTART, RESTART ACK, and STATUS ENQ messages at severity 4.
- ISDN SERV status audit messages for various triggers at different severities.

Additional ISDN Switch Types Supported for Network Emulation

ISDN enhancements introduced in Cisco IOS Release 12.3(1) extend network emulation capability to the Lucent 4ESS, 5ESS, and Nortel DMS-100 ISDN switch types. These switch types can be configured as network, but no additional changes were made and not all network-side features are supported.

How to Configure the ISDN Enhancements

This section contains the following procedures. Each procedure is optional and depends upon the settings required for your network.

- Configuring BCAC Service Audit Triggers, page 3 (optional)
- Configuring BCAC Service State Triggers, page 5 (optional)
- Configuring BCAC Message Retransmission, page 6 (optional)
- Configuring B-Channel Selection Order, page 7 (optional)
- Configuring ISDN Syslog Messages, page 8 (optional)

Configuring BCAC Service Audit Triggers

Perform this task to configure BCAC service audit triggers:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *port*:*channel*
4. **isdn bcac service audit**
5. **isdn bcac service audit trigger** *number*
6. **isdn bcac service audit interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *port*:*channel* **Example:** `Router(config)# interface serial 2:23`	Enters interface configuration mode on the specified serial port and channel.
Step 4	**isdn bcac service audit** **Example:** `Router(config-if)# isdn bcac service audit`	Enables BCAC service audits.
Step 5	**isdn bcac service audit trigger** *number* **Example:** `Router(config-if)# isdn bcac service audit trigger 2`	Enables individual BCAC service audit triggers.
Step 6	**isdn bcac service audit interface** **Example:** `Router(config-if)# isdn bcac service audit interface`	Specifies that BCAC service audits need to be triggered on the entire interface.

Examples

The following example shows how to enable service audits on serial interface 4:23:

```
interface serial 4:23
 isdn bcac service audit
```

The following example shows how to disable service trigger 4 on serial interface 4:23:

```
interface serial 4:23
 no isdn bcac service audit trigger 4
```

See the command page for the **isdn bcac service audit trigger** command for a list of the triggers that are set.

The following example shows how to configure service audits on the entire interface:

```
interface serial 4:23
 isdn bcac service audit interface
```

Configuring BCAC Service State Triggers

Perform this task to configure BCAC service state triggers:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *port*:*channel*
4. **isdn bcac service update provision**
5. **isdn bcac service update linkup**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *port*:*channel* **Example:** `Router(config)# interface serial 2:23`	Enters interface configuration mode on the specified serial port and channel.
Step 4	**isdn bcac service update provision** **Example:** `Router(config-if)# isdn bcac service update provision`	Enables BCAC service status functionality for provisioning the B channels.
Step 5	**isdn bcac service update linkup** **Example:** `Router(config-if)# isdn bcac service update linkup`	Triggers updates of the BCAC service states between peer nodes through exchange of SERV and SERV ACK messages.

Examples

The following example shows how to enable the SERV status message for provisioning the B channels on serial interface 4:23:

```
interface serial 4:23
 isdn bcac service update provision
```

The following example shows how to trigger service state updates on serial interface 4:23:

```
interface serial 4:23
 isdn bcac service update linkup
```

Configuring BCAC Message Retransmission

Perform this task to configure retransmission of BCAC service messages:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *port*:*channel*
4. **isdn bcac service timer** *timer-value*
5. **isdn bcac service retry max** *retries*
6. **isdn bcac service retry in-serv-on-fail**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *port*:*channel* **Example:** `Router(config)# interface serial 2:23`	Enters interface configuration mode on the specified serial port and channel.
Step 4	**isdn bcac service timer** *timer-value* **Example:** `Router(config-if)# isdn bcac service timer 600`	Changes the value of the BCAC T3M1 or T323 service message timer. • Valid range is from 500 to 120000 ms, and the default is 120000 ms.
Step 5	**isdn bcac service retry max** *retries* **Example:** `Router(config-if)# isdn bcac service retry max retries`	Specifies the maximum number of times a BCAC service message can be retransmitted when unacknowledged. • The default is 2 attempts, and you can enter a number from 0 to 127.

Step 6	`isdn bcac service retry in-serv-on-fail`
	
	Example:
	`Router(config-if)# isdn bcac service retry in-serv-on-fail`
		Specifies that the BCAC service state of the channel needs to be changed to In-Service, because no acknowledgment message was received.

Examples

The following example shows how to configure an option whereby, on service message exchange failure, the service state of the concerned channel or channels will be set to In-Service:

```
interface serial 2:23
 isdn bcac service retry in-serv-on-fail
```

The following example shows how to set the maximum number of service message retransmissions on serial interface 2:23 to 50:

```
interface serial 2:23
 isdn bcac service retry max 50
```

The following example shows how to change the service timers to 600 ms on serial interface 2:23:

```
interface serial 2:23
 isdn bcac service timer 600
```

Configuring B-Channel Selection Order

Perform this task to configure selection order of the ISDN B channels:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *port*:*channel*
4. **isdn bchan-number-order** {**ascending** | **descending**} [**round-robin**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface serial** *port*:*channel* **Example:** `Router(config)# interface serial 2:23`	Enters interface configuration mode on the specified serial port and channel.
Step 4	**isdn bchan-number-order** {**ascending** | **descending**} [**round-robin**] **Example:** `Router(config-if)# isdn bchan-number-order ascending round-robin`	Configures an ISDN PRI interface to make outgoing call selection in ascending or descending order. • The optional **round-robin** keyword adds round-robin selection functionality to the selection order.

Examples

The following example configures the outgoing B channel selection order on a PRI interface to be round-robin in ascending order:

```
interface serial 5:10
 isdn bchan-number-order ascending round-robin
```

Configuring ISDN Syslog Messages

Perform this task to configure logging of ISDN syslog messages:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **isdn logging**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**isdn logging** **Example:** `Router(config)# isdn logging`	Enables logging of ISDN syslog messages.

Examples

The following example shows how to configure ISDN syslog logging:

```
isdn logging
```

Configuration Examples for ISDN BCAC and Round-Robin Channel Selection Enhancements

See the examples following each task in the preceding sections, for ideas about how the ISDN CBAC enhancements and other new ISDN features can be introduced into your network.

Additional References

For additional information related to the ISDN enhancements, see the following sections:

Related Documents

Related Topic	Document Title
ISDN PRI configuration	Refer to the *"Configuring ISDN PRI" chapter in the "Signaling Configuration" part of the Cisco IOS Dial Technologies Configuration Guide*, Release 12.3.
ISDN PRI configuration commands	*Cisco IOS Dial Technologies Command Reference*, Release 12.3.
ISDN PRI configuration for voice, video, and fax	Refer to the chapter "Configuring ISDN Interfaces for Voice " in the *Cisco IOS Voice, Video, and Fax Configuration Guide*, Release 12.3.
ISDN PRI voice, video, and fax configuration commands	*Cisco IOS Voice, Video, and Fax Command Reference*, Release 12.3

Standards

Standards[1]	Title
AT&T PRI	Technical Report 41459–*AT&T ISDN Primary Rate Interface and Special Application Specification*; "User Network Interface Description," 1999.
National ISDN Council (NIC) PRI	SR (Special Report)-NWT-002343–*ISDN Primary Rate Interface Generic Guidelines for Customer Premises Equipment*, June 1993. SR-3887–*National ISDN Primary Rate Interface Customer Premises Equipment Generic Guidelines*, 1996.
Nortel PRI	NIS (Network Interface Specification)-A211-1–*DMS100 ISDN Primary Rate Network User Interface, 1993*.

1. Not all supported standards are listed.

MIBs

MIBs	MIBs Link
None	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 the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Dial Technologies Command Reference* at http://www.cisco.com/en/US/docs/ios/dial/command/reference/dia_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

New Commands

- **isdn bcac service audit**
- **isdn bcac service audit interface**
- **isdn bcac service audit trigger**
- **isdn bcac service retry in-serv-on-fail**
- **isdn bcac service retry max**
- **isdn bcac service timer**
- **isdn bcac service update linkup**
- **isdn bcac service update provision**
- **isdn logging**

Modified Commands

- **isdn bchan-number-order**
- **isdn protocol-emulate (dial)**

Glossary

PBX—private branch exchange.

RESTART—restart message.

RESTART ACK—restart acknowledge message.

STATUS ENQ—status enquiry message.

SERV—service message.

SERV ACK—service acknowledge message.

Note Refer to the *Internetworking Terms and Acronyms* for terms not included in this glossary.

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 1 shows the Cisco IOS keywords for the ISDN switch types and lists whether NFAS is supported.

Table 1 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 Snapshot Routing

This chapter describes how to configure snapshot routing. 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 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:

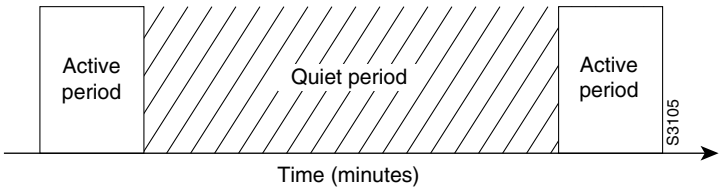
- Configuring static routes for dial-on-demand routing (DDR) interfaces
- Reducing the overhead of periodic updates sent by routing protocols to remote branch offices over a dedicated serial line

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.

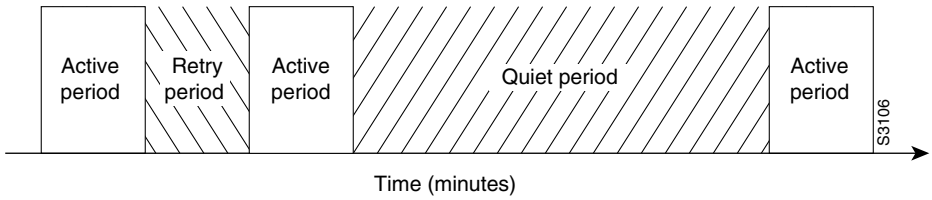
At the end of the active period, the router takes a snapshot of the entries in the routing table. These entries remain frozen during a quiet period. At the end of the quiet period, another active period starts during which routing information is again exchanged; see Figure 1.

Figure 1 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 2.

Figure 2 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.

Command	Purpose
`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
```

Reliable Static Routing Backup Using Object Tracking

First Published: May 10, 2001
Last Updated: November 20, 2009

The Reliable Static Routing Backup Using Object Tracking feature introduces the ability for the Cisco IOS software to use Internet Control Message Protocol (ICMP) pings to identify when a PPP over Ethernet (PPPoE) or IP Security Protocol (IPsec) Virtual Private Network (VPN) tunnel goes down, allowing the initiation of a backup connection from any alternative port. The Reliable Static Routing Backup Using Object Tracking feature is compatible with both preconfigured static routes and Dynamic Host Configuration Protocol (DHCP) configurations.

Finding Feature Information

For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Reliable Static Routing Backup Using Object Tracking" section on page 29.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Reliable Static Routing Backup Using Object Tracking

Dial-on-demand routing (DDR) must be configured if the backup connection is configured on a dialer interface. For more information on configuring DDR, refer to the "Dial-on-Demand Routing Configuration" part of the *Cisco IOS Dial Technologies Configuration Guide*.

Restrictions for Reliable Static Routing Backup Using Object Tracking

This feature is supported in all Cisco IOS software images for the Cisco 1700 series modular access routers except the Cisco IOS IP Base image.

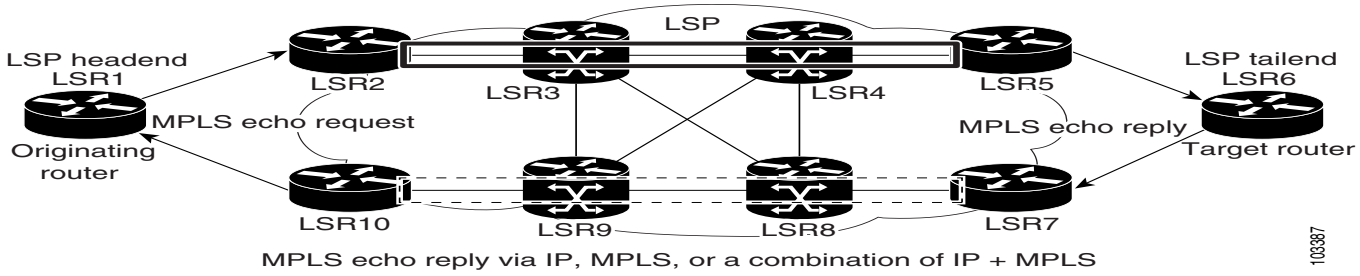
Information About Reliable Static Routing Backup Using Object Tracking

To configure the Reliable Static Routing Backup Using Object Tracking feature, you should understand the following concepts:

Reliable Static Routing Backup Using Object Tracking

The Reliable Static Routing Backup Using Object Tracking feature introduces the ability to reliably back up PPPoE or IPsec VPN deployments by initiating a DDR connection from an alternative port if the circuit to the primary gateway is interrupted. The Reliable Static Routing Backup Using Object Tracking feature can ensure reliable backup in the case of several catastrophic events, such as Internet circuit failure or peer device failure. A typical scenario is shown in Figure 1.

Figure 1 Reliable Static Routing Backup Using Object Tracking Scenario



Traffic from the remote LAN is forwarded to the main office from the primary interface of the remote router. If the connection to the main office is lost, the status of the tracked object changes from up to down. When the state of the tracked object changes to down, the routing table entry for the primary interface is removed and the preconfigured floating static route is installed on the secondary interface. Traffic is then forwarded to the preconfigured destination from the secondary interface. If DDR is configured on the secondary interface, interesting traffic will trigger DDR. The backup circuit can be configured to use the public switched telephone network (PSTN) or the Internet. When the state of the tracked object changes from down to up, the routing table entry for the primary interface is reinstalled and the floating static route for the secondary interface is removed.

Cisco IOS IP SLAs

The Reliable Static Routing Backup Using Object Tracking feature uses Cisco IOS IP Service Level Agreements (IP SLAs), a network monitoring feature set, to generate ICMP pings to monitor the state of the connection to the primary gateway. Cisco IOS IP SLAs is configured to ping a target, such as a publicly routable IP address or a target inside the corporate network. The pings are routed from the primary interface only. A track object is created to monitor the status of the Cisco IOS IP SLAs configuration. The track object informs the client, the static route, if a state change occurs. The preconfigured floating static route on the secondary interface will be installed when the state changes from up to down.

HTTP GET, User Datagram Protocol (UDP) echo, or any other protocol supported by Cisco IOS IP SLAs can be used instead of ICMP pings.

Benefits of Reliable Static Routing Backup Using Object Tracking

PPPoE and IPsec VPN deployments provide cost-effective and secure Internet-based solutions that can replace traditional dialup and Frame Relay circuits.

The Reliable Static Routing Backup Using Object Tracking feature can determine the state of the primary connection without enabling a dynamic routing protocol.

The Reliable Static Routing Backup Using Object Tracking feature introduces a reliable backup solution for PPPoE and IPsec VPN deployments, allowing these solutions to be used for critical circuits that must not go down without a backup circuit automatically engaging.

How to Configure Reliable Static Routing Backup Using Object Tracking

This section contains the following tasks:

- Configuring the Primary Interface for Reliable Static Routing Backup Using Object Tracking, page 4 (required)
- Configuring the Backup Interface for Reliable Static Routing Backup Using Object Tracking, page 8
- Configuring Network Monitoring with Cisco IOS IP SLAs for Reliable Static Routing Backup Using Object Tracking, page 9 (required)
- Configuring the Routing Policy for Reliable Static Routing Backup Using Object Tracking, page 15 (required)
- Configuring the Default Route for the Primary Interface Using Static Routing, page 22 (required)
- Configuring a Floating Static Default Route on the Secondary Interface, page 22 (required)
- Verifying the State of the Tracked Object for Reliable Static Routing Backup Using Object Tracking, page 23 (optional)

Configuring the Primary Interface for Reliable Static Routing Backup Using Object Tracking

You must configure the connection between the primary interface and the remote gateway. The status of this connection will be monitored by the Reliable Static Routing Backup Using Object Tracking feature.

The primary interface can be configured in one of three ways: for PPPoE, DHCP, or static routing. You must choose one of these configuration types. If you are not sure of which method to use with your network configuration, consult your Internet service provider (ISP) or network administrator.

Perform one of the following tasks to configure the primary interface:

- Configuring the Primary Interface for PPPoE, page 4
- Configuring the Primary Interface for DHCP, page 5
- Configuring the Primary Interface for Static Routing, page 7

Configuring the Primary Interface for PPPoE

Perform this task to configure the primary interface for PPPoE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **description** *string*
5. **no ip address**
6. **pvc** [**name**] *vpi*/*vci* [**ces** | **ilmi** | **qsaal** | **smds** | **l2transport**]
7. **pppoe-client dial-pool-number** *number* [**dial-on-demand**]

8. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface ATM 2/0`	Configures an interface type and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Router(config-if)# description primary-link`	Adds a description to the interface configuration.
Step 5	**no ip address** **Example:** `Router(config-if)# no ip address`	Removes IP addresses configured on the interface.
Step 6	**pvc** [**name**] *vpi*/*vci* [**ces** | **ilmi** | **qsaal** | **smds** | **l2transport**] **Example:** `Router(config-if)# pvc 0/33`	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.
Step 7	**pppoe-client dial-pool-number** *number* [**dial-on-demand**] **Example:** `Router(config-if-atm-vc)# pppoe-client dial-pool-number 1`	Configures a PPPoE client and specifies DDR functionality.
Step 8	**exit** **Example:** `Router(config-if-atm-vc)# exit`	Exits ATM virtual circuit configuration mode.

Configuring the Primary Interface for DHCP

Perform this task to configure the primary interface for DHCP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **description** *string*
5. **ip dhcp client route track** *number*
6. **ip address dhcp**
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface ethernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Router(config-if)# description primary-link`	Adds a description to the interface configuration.
Step 5	**ip dhcp client route track** *number* **Example:** `Router(config-if)# ip dhcp client route track 123`	Configures the DHCP client to associate any added routes with the specified track number. • **route track** *number*—Associates a track object with the DHCP-installed static route. Valid values for the *number* argument range from 1 to 500. **Note** You must configure the **ip dhcp client** command before issuing the **ip address dhcp** command on an interface. The **ip dhcp client** command is checked only when an IP address is acquired from DHCP. If the **ip dhcp client** command is issued after an IP address has been acquired from DHCP, it will not take effect until the next time the router acquires an IP address from DHCP.

	Command or Action	Purpose
Step 6	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Acquires an IP address on an Ethernet interface from DHCP.
Step 7	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Configuring the Primary Interface for Static Routing

Perform this task to configure the primary interface for static routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **description** *string*
5. **ip address** *ip-address mask* [**secondary**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface FastEthernet 0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**description** *string* **Example:** `Router(config-if)# description primary-link`	Adds a description to the interface configuration.

	Command or Action	Purpose
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.200.225 255.0.0.0`	Sets a primary or secondary IP address for an interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Configuring the Backup Interface for Reliable Static Routing Backup Using Object Tracking

You must configure a backup interface to contact the remote gateway. If the connection between the primary interface and the remote gateway goes down, the backup interface will be used.

Perform the following task to configure the backup interface. This task applies to PPPoE, DHCP, and static routing configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number* [*name-tag*]
4. **description** *string*
5. **ip address** *ip-address mask* [**secondary**]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* [*name-tag*] **Example:** `Router(config)# interface Dialer 0`	Configures an interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**description** *string* **Example:** `Router(config-if)# description backup-link`	Adds a description to an interface configuration.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 209.165.201.1 255.255.255.0`	Sets a secondary IP address for an interface. **Note** If the connection on the primary interface goes down, the secondary interface is used as a backup interface.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.

Configuring Network Monitoring with Cisco IOS IP SLAs for Reliable Static Routing Backup Using Object Tracking

The Reliable Static Routing Backup Using Object Tracking feature uses a Cisco IOS IP SLAs configuration to generate ICMP pings to monitor the state of the connection to the primary gateway.

Beginning in Cisco IOS Release 12.3(14)T, the command used to configure Cisco IOS IP SLAs was modified.

Perform one of the following tasks to configure Cisco IOS IP SLAs depending on which Cisco IOS software release you are running:

- Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.3(8)T, 12.3(11)T, 12.2(33)SRA, and 12.2(33)SRE, page 9
- Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.3(14)T, 12.4, 12.4(2)T, and 12.2(33)SXH, page 11
- Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.4(4)T, 15.(0)1M, and Later Releases, page 13

Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.3(8)T, 12.3(11)T, 12.2(33)SRA, and 12.2(33)SRE

Perform this task to create Cisco IOS IP SLAs depending on which Cisco IOS software release you are running. This task applies to PPPoE, DHCP, and static routing configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **rtr** [*operation-number*]
4. **type echo protocol ipIcmpEcho** {*destination-ip-address* | *destination-hostname*} [**source-ipaddr** {*ip-address* | *hostname*}]
5. **timeout** *milliseconds*

6. **frequency** *seconds*
7. **threshold** *milliseconds*
8. **exit**
9. **rtr schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*]
10. **track** *object-number* **rtr** *rtr-operation* {**state** | **reachability**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**rtr** [*operation-number*] **Example:** `Router(config)# rtr 1`	Begins configuration for a Cisco IOS IP SLAs operation and enters RTR configuration mode.
Step 4	**type echo protocol ipIcmpEcho** {*destination-ip-address* | *destination-hostname*} [**source-ipaddr** {*ip-address* | *hostname*}] **Example:** `Router(config-rtr)# type echo protocol ipIcmpEcho 172.16.23.7`	Configures a Cisco IOS IP SLAs end-to-end echo response time probe operation.
Step 5	**timeout** *milliseconds* **Example:** `Router(config-rtr)# timeout 1000`	Sets the amount of time for which the Cisco IOS IP SLAs operation waits for a response from its request packet.
Step 6	**frequency** *seconds* **Example:** `Router(config-rtr)# frequency 3`	Sets the rate at which a specified Cisco IOS IP SLAs operation is sent into the network.
Step 7	**threshold** *milliseconds* **Example:** `Router(config-rtr)# threshold 2`	Sets the rising threshold (hysteresis) that generates a reaction event and stores history information for the Cisco IOS IP SLAs operation.
Step 8	**exit** **Example:** `Router(config-rtr)# exit`	Exits RTR configuration mode.

	Command or Action	Purpose
Step 9	**rtr schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] **Example:** `Router(config)# rtr schedule 1 life forever start-time now`	Configures a Cisco IOS IP SLAs ICMP echo operation.
Step 10	**track** *object-number* **rtr** *rtr-operation* {**state** | **reachability**} **Example:** `Router(config)# track 123 rtr 1 reachability`	Tracks the state of a Cisco IOS IP SLAs operation and enters tracking configuration mode.
Step 11	**end** **Example:** `Router(config-track-list)# end`	Exits tracking configuration mode.

Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.3(14)T, 12.4, 12.4(2)T, and 12.2(33)SXH

Perform this task to create an Cisco IP SLAs configuration to ping the target address depending on which Cisco IOS software release you are running. This task applies to PPPoE, DHCP, and static routing configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla monitor** [*operation-number*]
4. **type echo protocol ipIcmpEcho** {*destination-ip-address* | *destination-hostname*} [**source-ipaddr** {*ip-address* | *hostname*} | **source-interface** *interface-name*]
5. **timeout** *milliseconds*
6. **frequency** *seconds*
7. **threshold** *milliseconds*
8. **exit**
9. **ip sla monitor schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
10. **track** *object-number* **rtr** *rtr-operation* {**state** | **reachability**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip sla monitor** [*operation-number*] **Example:** `Router(config)# ip sla monitor 1`	Begins configuring a Cisco IOS IP SLAs operation and enters IP SLA monitor configuration mode.
Step 4	**type echo protocol ipIcmpEcho** {*destination-ip-address* | *destination-hostname*} [**source-ipaddr** {*ip-address* | *hostname*} | **source-interface** *interface-name*] **Example:** `Router(config-sla-monitor)# type echo protocol ipIcmpEcho 172.16.23.7`	Configures a Cisco IOS IP SLAs end-to-end ICMP echo response time operation and enters IP SLAs ICMP echo configuration mode.
Step 5	**timeout** *milliseconds* **Example:** `Router(config-sla-monitor-echo)# timeout 1000`	Sets the amount of time for which the Cisco IOS IP SLAs operation waits for a response from its request packet.
Step 6	**frequency** *seconds* **Example:** `Router(config-sla-monitor-echo)# frequency 3`	Sets the rate at which a specified Cisco IOS IP SLAs operation is sent into the network.
Step 7	**threshold** *milliseconds* **Example:** `Router(config-sla-monitor-echo)# threshold 2`	Sets the rising threshold (hysteresis) that generates a reaction event and stores history information for the Cisco IOS IP SLAs operation.
Step 8	**exit** **Example:** `Router(config-sla-monitor-echo)# exit`	Exits IP SLAs ICMP echo configuration mode.
Step 9	**ip sla monitor schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [recurring] **Example:** `Router(config)# ip sla monitor schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLAs operation.

	Command or Action	Purpose
Step 10	**track** *object-number* **rtr** *rtr-operation* {**state** | **reachability**} **Example:** `Router(config)# track 123 rtr 1 reachability`	Tracks the state of a Cisco IOS IP SLAs operation and enters tracking configuration mode.
Step 11	**end** **Example:** `Router(config-track-list)# end`	Exits tracking configuration mode.

Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.4(4)T, 15.(0)1M, and Later Releases

Perform this task to create Cisco IP SLAs configuration in Cisco IOS Release 12.4(4)T, 15.0(1)M, and later releases to ping the target address. This task applies to PPPoE, DHCP, and static routing configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip sla** [*operation-number*]
4. **icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-name*]
5. **timeout** *milliseconds*
6. **frequency** *seconds*
7. **threshold** *milliseconds*
8. **exit**
9. **ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [**recurring**]
10. **track** *object-number* **rtr** *rtr-operation* {**state** | **reachability**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**ip sla** [*operation-number*] **Example:** `Router(config)# ip sla 1`	Begins configuring a Cisco IOS IP SLAs operation and enters IP SLA configuration mode.
Step 4	**icmp-echo** {*destination-ip-address* | *destination-hostname*} [**source-ip** {*ip-address* | *hostname*} | **source-interface** *interface-name*] **Example:** `Router(config-ip-sla)# icmp-echo 172.16.23.7`	Configures a Cisco IOS IP SLAs end-to-end ICMP echo response time operation and enters IP SLAs ICMP echo configuration mode.
Step 5	**timeout** *milliseconds* **Example:** `Router(config-ip-sla-echo)# timeout 1000`	Sets the amount of time for which the Cisco IOS IP SLAs operation waits for a response from its request packet.
Step 6	**frequency** *seconds* **Example:** `Router(config-ip-sla-echo)# frequency 3`	Sets the rate at which a specified Cisco IOS IP SLAs operation is sent into the network.
Step 7	**threshold** *milliseconds* **Example:** `Router(config-ip-sla-echo)# threshold 2`	Sets the rising threshold (hysteresis) that generates a reaction event and stores history information for the Cisco IOS IP SLAs operation.
Step 8	**exit** **Example:** `Router(config-ip-sla-echo)# exit`	Exits IP SLAs ICMP echo configuration mode.
Step 9	**ip sla schedule** *operation-number* [**life** {**forever** | *seconds*}] [**start-time** {*hh*:*mm*[:*ss*] [*month day* | *day month*] | **pending** | **now** | **after** *hh*:*mm*:*ss*}] [**ageout** *seconds*] [recurring] **Example:** `Router(config-ip-sla-echo)# ip sla schedule 1 life forever start-time now`	Configures the scheduling parameters for a single Cisco IOS IP SLAs operation.
Step 10	**track** *object-number* **rtr** *rtr-operation* {**state** | **reachability**} **Example:** `Router(config)# track 123 rtr 1 reachability`	Tracks the state of a Cisco IOS IP SLAs operation and enters tracking configuration mode.
Step 11	**end** **Example:** `Router(config-track-list)# end`	Exits tracking configuration mode.

Configuring the Routing Policy for Reliable Static Routing Backup Using Object Tracking

In order to track the status of the primary connection to the remote gateway, the Cisco IOS IP SLAs ICMP pings must be routed only from the primary interface.

Perform one of the following tasks to configure a routing policy that will ensure that the Cisco IOS IP SLAs pings are always routed out of the primary interface:

- Configuring a Routing Policy for PPPoE, page 15
- Configuring a Routing Policy for DHCP, page 17
- Configuring a Routing Policy for Static Routing, page 18

Configuring a Routing Policy for PPPoE

Perform this task to configure a routing policy if the primary interface is configured for PPPoE.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
6. **set interface** *type number* [*... type number*]
7. **exit**
8. **ip local policy route-map** *map-tag*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 101 permit icmp any host 172.16.23.7 echo`	Defines an extended IP access list.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map MY-LOCAL-POLICY permit 10`	Enters route-map configuration mode and defines the conditions for redistributing routes from one routing protocol into another.
Step 5	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Router(config-route-map)# match ip address 101`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
Step 6	**set interface** *type number* [*... type number*] **Example:** `Router(config-route-map)# set interface null 0`	Indicates where to output packets that pass a match clause of a route map for policy routing. **Note** The interface must be configured for **null 0** in this scenario. If the next hop is not set because the interface is down, the packet is routed to the null interface and discarded. Otherwise policy routing fails and the packet is routed using the Routing Information Base (RIB) card. Routing the packet using the RIB card is undesirable.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode.
Step 8	**ip local policy route-map** *map-tag* **Example:** `Router(config)# ip local policy route-map MY-LOCAL-POLICY`	Identifies a route map to use for local policy routing.
Step 9	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Configuring a Routing Policy for DHCP

Perform this task to ensure that the primary interface is configured for DHCP:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
6. **set ip next-hop dynamic dhcp**
7. **exit**
8. **ip local policy route-map** *map-tag*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 101 permit icmp any host 172.16.23.7 echo`	Defines an extended IP access list.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map MY-LOCAL-POLICY permit 10`	Enters route-map configuration mode and defines the conditions for redistributing routes from one routing protocol into another.

	Command or Action	Purpose
Step 5	**match ip address** {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*] **Example:** `Router(config-route-map)# match ip address 101`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
Step 6	**set ip next-hop dynamic dhcp** **Example:** `Router(config-route-map)# set ip next-hop dynamic dhcp`	Sets the next hop to the gateway that was most recently learned by the DHCP client.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode.
Step 8	**ip local policy route-map** *map-tag* **Example:** `Router(config)# ip local policy route-map MY-LOCAL-POLICY`	Identifies a route map to use for local policy routing.
Step 9	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Configuring a Routing Policy for Static Routing

Perform one of the following tasks if the primary interface is configured for static routing:

- Configuring a Routing Policy for Static Routing with a Point-to-Point Primary Gateway, page 18
- Configuring a Routing Policy for Static Routing with a Multipoint Primary Gateway, page 20

Configuring a Routing Policy for Static Routing with a Point-to-Point Primary Gateway

Perform this task to configure a routing policy if the primary interface is configured for static routing and the primary gateway is a point-to-point gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* | *access-list-name*} [... *access-list-number* | ... *access-list-name*]

6. **set interface** *type number* [*... type number*]
7. **exit**
8. **ip local policy route-map** *map-tag*
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 101 permit icmp any host 172.16.23.7 echo`	Defines an extended IP access list.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map MY-LOCAL-POLICY permit 10`	Enters route-map configuration mode and defines the conditions for redistributing routes from one routing protocol into another.
Step 5	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Router(config-route-map)# match ip address 101`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
Step 6	**set interface** *type number* [*...type number*] **Example:** `Router(config-route-map)# set interface dialer 0 Null 0`	Indicates where to output packets that pass a match clause of a route map for policy routing.
Step 7	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode.

	Command or Action	Purpose
Step 8	**ip local policy route-map** *map-tag* **Example:** `Router(config)# ip local policy route-map MY-LOCAL-POLICY`	Identifies a route map to use for local policy routing.
Step 9	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Configuring a Routing Policy for Static Routing with a Multipoint Primary Gateway

Perform this task to configure a routing policy if the primary interface is configured for static routing and the primary gateway is a multipoint gateway.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
4. **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]
5. **match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*]
6. **set ip next-hop** *ip-address* [*... ip-address*]
7. **set interface** *type number* [*... type number*]
8. **exit**
9. **ip local policy route-map** *map-tag*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **icmp** *source source-wildcard destination destination-wildcard* [*icmp-type* [*icmp-code*] | *icmp-message*] [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**] **Example:** `Router(config)# access-list 101 permit icmp any host 172.16.23.7 echo`	Defines an extended IP access list.
Step 4	**route-map** *map-tag* [**permit** | **deny**] [*sequence-number*] **Example:** `Router(config)# route-map MY-LOCAL-POLICY permit 10`	Enters route-map configuration mode and defines the conditions for redistributing routes from one routing protocol into another.
Step 5	**match ip address** {*access-list-number* | *access-list-name*} [*... access-list-number* | *... access-list-name*] **Example:** `Router(config-route-map)# match ip address 101`	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
Step 6	**set ip next-hop** *ip-address* [*... ip-address*] **Example:** `Router(config-route-map)# set ip next-hop 10.1.1.242`	Indicates where to output packets that pass a match clause of a route map for policy routing.
Step 7	**set interface** *type number* [*... type number*] **Example:** `Router(config-route-map)# set interface null 0`	Indicates where to output packets that pass a match clause of a route map for policy routing.
Step 8	**exit** **Example:** `Router(config-route-map)# exit`	Exits route-map configuration mode.
Step 9	**ip local policy route-map** *map-tag* **Example:** `Router(config)# ip local policy route-map MY-LOCAL-POLICY`	Identifies a route map to use for local policy routing.
Step 10	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Configuring the Default Route for the Primary Interface Using Static Routing

Perform this task to configure the static default route only if you are using static routing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *prefix mask* {*ip-address* | *interface-type interface-number* [*ip-address*]} [*distance*] [*name*] [**permanent** | **track** *number*] [**tag** *tag*] **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.242 track 123`	Establishes static routes. • **track** *number*—Specifies that the static route will be installed only if the configured track object is up.
Step 4	**end** **Example:** `Router(config)# end`	Exits global configuration mode.

Configuring a Floating Static Default Route on the Secondary Interface

Perform this task to configure a floating static default route on the secondary interface. This task applies to PPPoE, DHCP, and static routing configurations.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip route** *network-number network-mask* {*ip-address* | *interface*} [*distance*] [**name** *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip route** *network-number network-mask* {*ip-address* | *interface*} [*distance*] [**name** *name*] **Example:** `Router(config)# ip route 0.0.0.0 0.0.0.0 10.2.2.125 254`	Establishes static routes and defines the next hop.

Verifying the State of the Tracked Object for Reliable Static Routing Backup Using Object Tracking

Perform the following task to determine if the state of the tracked object is up or down.

SUMMARY STEPS

1. **enable**
2. **show ip route track-table**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip route track-table** **Example:** `Router# show ip route track-table`	Displays information about the IP route track table.

Configuration Examples for Reliable Static Routing Backup Using Object Tracking

This section provides the following configuration examples:

Configuring Reliable Static Routing Backup Using Object Tracking Using PPPoE: Example

The following example shows how to configure the Reliable Static Routing Backup Using Object Tracking feature using PPPoE. The primary interface is an ATM interface, and the backup interface is a BRI interface. This example applies to Cisco IOS Release 12.3(8)T, 12.3(11)T, 12.2(33)SRA, 12.2(33)SXH, and 12.2(33)SRE.

```
interface ATM 0
 description primary-link
 no ip address
 pvc 0/33
  pppoe-client dial-pool-number 1
!
interface BRI 0
 description backup-link
 ip address 10.2.2.2 255.0.0.0
!
rtr 1
 type echo protocol ipIcmpEcho 172.16.23.7
 timeout 1000
 frequency 3
 threshold 2

rtr schedule 1 life forever start-time now
track 123 rtr 1 reachability

access list 101 permit icmp any host 172.16.23.7 echo
route map MY-LOCAL-POLICY permit 10
 match ip address 101
 set interface null 0
!
ip local policy route-map MY-LOCAL-POLICY

ip route 0.0.0.0 0.0.0.0 10.2.2.125 254
```

Configuring Reliable Static Routing Backup Using Object Tracking Using DHCP: Example

The following example show how to configure the Reliable Static Routing Backup Using Object Tracking feature using DHCP. The primary interface is an Ethernet interface, and the backup interface is a serial interface. This example applies to Cisco IOS Release 12.3(14)T.

```
!
ip dhcp-client default-router distance 25
ip sla monitor 1
  type echo protocol ipIcmpEcho 172.16.23.7
  timeout 1000
  threshold 2
  frequency 3
ip sla monitor schedule 1 life forever start-time now
track 123 rtr 1 reachability
!
interface Ethernet0/0
  description primary-link
  ip dhcp client route track 123
  ip address dhcp
!
interface Serial2/0
  description backup-link
  ip address 209.165.202.129 255.255.255.255
!
ip local policy route-map MY-LOCAL-POLICY
ip route 0.0.0.0 0.0.0.0 10.2.2.125 254
!
access-list 101 permit icmp any host 172.16.23.7 echo
route-map MY-LOCAL-POLICY permit 10
  match ip address 101
  set ip next-hop dynamic dhcp
```

Configuring Reliable Static Routing Backup Using Object Tracking: Example

The following example shows how to configure the Reliable Static Routing Backup Using Object Tracking feature using static routing for a point-to-point primary gateway. The primary interface is a PPPoE Fast Ethernet interface, and the backup interface is a dialer interface. This example applies to Cisco IOS Release 12.3(14)T and later releases.

```
interface FastEthernet 0/0
 description primary-link
 ip address 209.165.202.129 255.255.255.255

interface Dialer 0
 description backup-link
 ip address 209.165.200.225 255.255.255.255

ip sla monitor 1
 type echo protocol ipIcmpEcho 172.16.23.7
 timeout 1000
 frequency 3
 threshold 2

ip sla monitor schedule 1 life forever start-time now
track 123 rtr 1 reachability
```

```
access list 101 permit icmp any host 172.16.23.7 echo
route map MY-LOCAL-POLICY permit 10
 match ip address 101
 set interface dialer 0 null 0
!
ip local policy route-map MY-LOCAL-POLICY

ip route 0.0.0.0 0.0.0.0 10.1.1.242 track 123
ip route 0.0.0.0 0.0.0.0 10.2.2.125 254
```

The following example configures the Reliable Static Routing Backup Using Object Tracking feature using static routing for a multipoint primary gateway. Both the primary interface and the backup interface are Ethernet interfaces. This example applies to Cisco IOS Release 12.3(14)T and later releases.

```
interface ethernet 0
 description primary-link
 ip address 209.165.202.129 255.255.255.255

interface ethernet 1
 description backup-link
 ip address 209.165.200.225 255.255.255.255

ip sla monitor 1
 type echo protocol ipIcmpEcho 172.16.23.7
 timeout 1000
 frequency 3
 threshold 2

ip sla monitor schedule 1 life forever start-time now
track 123 rtr 1 reachability

access list 101 permit icmp any host 172.16.23.7 echo
route map MY-LOCAL-POLICY permit 10
 match ip address 101
 set ip next-hop 10.1.1.242
 set interface null 0
!
ip local policy route-map MY-LOCAL-POLICY

ip route 0.0.0.0 0.0.0.0 10.1.1.242 track 123
ip route 0.0.0.0 0.0.0.0 10.2.2.125 254
```

Verifying the State of the Tracked Object: Example

The following example displays information about track objects in the IP route track table:

```
Router# show ip route track-table

ip route 0.0.0.0 0.0.0.0 10.1.1.242 track-object 123 state is [up]
```

Additional References

The following sections provide references related to the Reliable Static Routing Backup Using Object Tracking feature.

Related Documents

Related Topic	Document Title
IPsec configuration tasks	"IP Security VPN Monitoring" module in the *Cisco IOS Security Configuration Guide*
IPsec commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Security Command Reference*
VPDN configuration tasks	"Configuring AAA for VPDN" module in the *Cisco IOS VPDN Configuration Guide*
VPDN commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS VPDN Command Reference*
ATM virtual circuit bundles	"ATM RBE" module in the *Cisco IOS Broadband Access Aggregation and DSL Configuration Guide*
PPPoE commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Wide-Area Networking Command Reference*
Dial access specialized features	"Dial Access Specialized Features" module in the *Cisco IOS Dial Technologies Configuration Guide*
DDR commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Dial Technologies Command Reference*
IP SLAs configuration tasks	"IP SLAs" module in the *Cisco IOS IP SLAs Configuration Guide*
IP SLAs commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS IP SLAs Command Reference*

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Reliable Static Routing Backup Using Object Tracking

Table 1 lists the features in this module and provides links to specific configuration information. Only features that were introduced or modified in Cisco IOS Release 12.2(33)SX or Cisco IOS Releases 12.2(33)SRE or 15.0(1)M or a later release appear in this table.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Table 1 Feature Information for Reliable Static Routing Backup Using Object Tracking

Feature Name	Releases	Feature Information
Reliable Static Routing Backup Using Object Tracking	12.2(33)SXH 12.2(33)SRA 12.2(33)SRE 12.3(8)T 12.3(14)T 15.0(1)M	The Reliable Static Routing Backup Using Object Tracking feature introduces the ability for the Cisco IOS software to use ICMP pings to identify when a PPPoE or IPsec VPN tunnel goes down, allowing the initiation of a backup connection from any alternative port. The Reliable Static Routing Backup Using Object Tracking feature is compatible with both preconfigured static routes and DHCP configurations. The following sections provide information about this feature: • Information About Reliable Static Routing Backup Using Object Tracking, page 2 • How to Configure Reliable Static Routing Backup Using Object Tracking, page 4 The following commands were introduced or modified: **ip dhcp client route**, **ip route prefix mask**, **set ip next-hop dynamic**, and **show ip route track-table**.

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
```

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:

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.
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
```

Dialer Watch Connect Delay

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the Dialer Watch Connect Delay feature in Cisco IOS Release 12.2(8)T and includes the following sections:

Feature Overview

The Dialer Watch Connect Delay feature introduces the ability to configure a delay in bringing up a secondary link when a primary link that is monitored by Dialer Watch goes down and is removed from the routing table. Previously, the router would instantly dial a secondary route without allowing time for the primary route to come back up. When the Dialer Watch Connect Delay feature is configured, the router will check for availability of the primary link at the end of the specified delay time before dialing the secondary link.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Benefits

The Dialer Watch Connect Delay feature allows users greater control over the use of a secondary link on monitored IP addresses or networks. Configuring the router to delay bringing up a secondary link when the watched primary link goes down will allow time for the primary link to be restored in the event of a temporary outage.

Related Documents

- The part "Dial-on-Demand Routing Configuration" in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.
- The chapter "Configuring Dial Backup Using Dialer Watch" in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.
- *Cisco IOS Dial Technologies Command Reference*, Release 12.2

Supported Platforms

- Cisco 805
- Cisco 806
- Cisco 820
- Cisco 827
- Cisco 828
- Cisco 1600 series
- Cisco 1700
- Cisco 1710
- Cisco 2600 series
- Cisco 3640
- Cisco 3660
- Cisco 7100
- Cisco 7200
- Cisco 7500
- soho 78
- mc3810
- C6MSFC2

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to quickly determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

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

Supported Standards, MIBs, and RFCs

Standards

No new or modified standards are supported by this feature.

MIBs

No new or modified MIBs are supported by this feature.

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 the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

No new or modified RFCs are supported by this feature.

Prerequisites

Dial-on-Demand routing (DDR) must be configured and Dialer Watch must be enabled. For more information on configuring DDR, refer to the following documents:

- The part “Dial-on-Demand Routing Configuration” in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.
- The chapter “Configuring Dial Backup Using Dialer Watch” in the *Cisco IOS Dial Technologies Configuration Guide,* Release 12.2.

Configuration Tasks

See the following sections for configuration tasks for the Dialer Watch Connect Delay feature. Each task in the list is identified as either required or optional.

- Configuring a Delay Before Activating a Secondary Link (required)

- Configuring a Delay Before Disconnecting the Secondary Link (optional)

Configuring a Delay Before Activating a Secondary Link

To configure the router to delay before dialing a secondary link when the primary link goes down, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer watch-list` *group-number* `delay connect` *connect-time*	Configures a delay in dialing the secondary link when the primary link becomes unavailable. • The **delay connect** keyword phrase specifies that the router will delay dialing the secondary link when the primary link becomes unavailable. • The *connect-time* argument specifies the 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.

Configuring a Delay Before Disconnecting the Secondary Link

To configure the router to delay before disconnecting a secondary link when the primary link is reestablished, use the following command in global configuration mode:

Command	Purpose
`Router(config)# dialer watch-list` *group-number* `delay disconnect` *disconnect-time*	Configures a delay in disconnecting the secondary link after detecting availability of the primary link. • The **delay disconnect** keyword phrase specifies that the router will delay disconnecting the secondary link after detecting availability of the primary link. • The *disconnect-time* argument specifies the time, in seconds, after which the router disconnects the secondary link once the primary link has been detected. Valid times range from 1 to 2147483.

Verifying Dialer Watch Connect Delay Configuration

To verify the cofiguration fo the Dialer Watch Connect Delay feature, perform the following steps:

Step 1 Enter the **show running-config** command to verify the configuration of Dialer Watch connect and disconnect delays:

```
router# show running-config

dialer watch-list 1 ip 10.1.1.1 255.0.0.0
dialer watch-list 1 delay connect 20
dialer watch-list 1 delay disconnect 20
```

```
dialer-list 1 protocol ip permit
```

Step 2 Enter the **debug dialer** command:

```
router# debug dialer

Connect Delay
-----------------------

*Mar  1 04:29:16:DDR:Dialer Watch:watch-group = 1
*Mar  1 04:29:16:DDR:   network 5.0.0.0/255.0.0.0 DOWN,
*Mar  1 04:29:16:DDR:   network 4.0.0.0/255.0.0.0 DOWN,
*Mar  1 04:29:16:DDR:   network 3.0.0.0/255.0.0.0 DOWN,
*Mar  1 04:29:16:DDR:   primary DOWN
*Mar  1 04:29:16:DDR:Dialer Watch: Primary of group 1 DOWN - start dial-backup timer


Disconnect delay
-----------------------

*Mar  1 04:31:11:BR2/0:1 DDR:idle timeout
*Mar  1 04:31:11:DDR:Dialer Watch:watch-group = 1
*Mar  1 04:31:11:DDR:   network 5.0.0.0/255.0.0.0 UP,
*Mar  1 04:31:11:DDR:   primary UP
*Mar  1 04:31:11:BR2/0:1 DDR:starting watch disconnect timer
*Mar  1 04:31:46:BR2/0:1 DDR:watch disconnect timeout
*Mar  1 04:31:46:DDR:Dialer Watch:watch-group = 1
*Mar  1 04:31:46:DDR:   network 5.0.0.0/255.0.0.0 UP,
*Mar  1 04:31:46:DDR:   primary UP
```

Configuration Examples

This section provides the following configuration examples:

- Configuring a Delay Before Activating a Secondary Link Example
- Configuring a Delay Before Disconnecting a Secondary Link Example

Configuring a Delay Before Activating a Secondary Link Example

The following example configures the router to wait 10 seconds before verifying that the primary link is still down and dialing a secondary link:

```
dialer watch-list 1 ip 10.1.1.0 255.255.255.0
dialer watch-list 1 delay connect 10
```

Configuring a Delay Before Disconnecting a Secondary Link Example

The following example configures the router to wait 10 seconds to disconnect a secondary link once the primary link has been reestablished:

```
dialer watch-list 1 ip 10.1.1.0 255.255.255.0
dialer watch-list 1 delay disconnect 10
```

Command Reference

The following new command is pertinent to this feature. To see the command pages for this command and other commands used with this feature, go to the *Cisco IOS Master Commands List*, Release 12.4, at http://www.cisco.com/univercd/cc/td/doc/product/software/ios124/124mindx/124index.htm.

- **dialer watch-list delay**

Configuring Cisco Easy IP

This chapter describes how to configure the Cisco Easy IP 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 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.

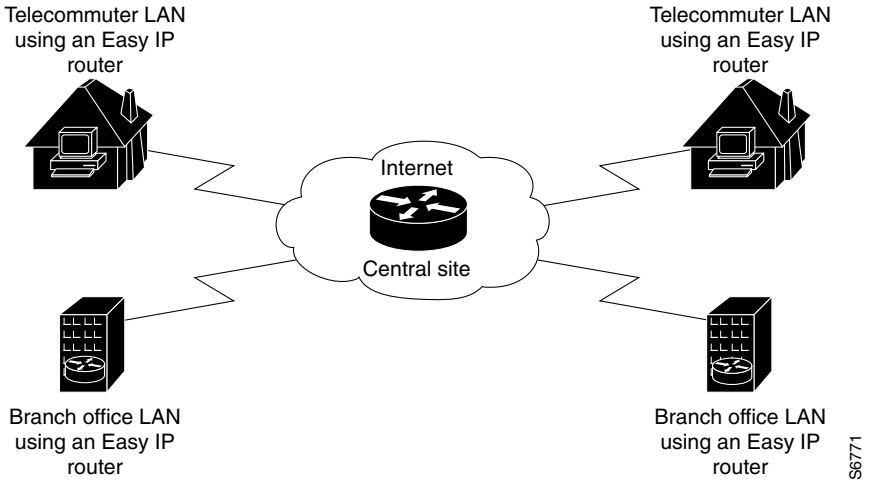
Cisco Easy IP provides the following benefits:

- 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 1 shows a typical scenario for using the Cisco Easy IP feature.

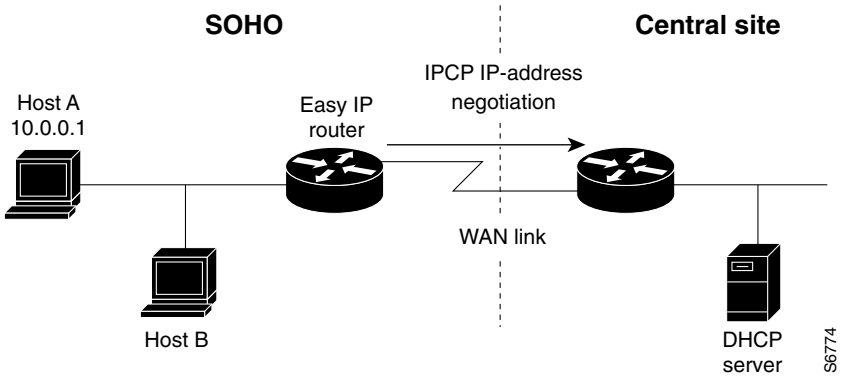
Figure 1 Telecommuter and Branch Office LANs Using Cisco Easy IP



Steps 1 through 4 show how Cisco Easy IP works:

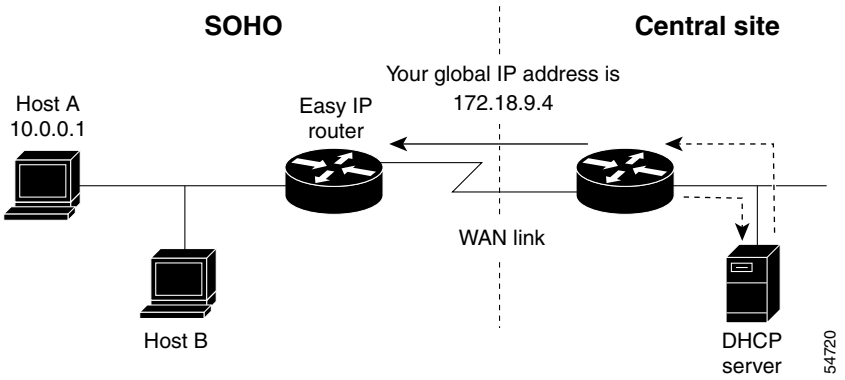
Step 1 When a SOHO host generates "interesting" traffic (as defined by Access Control Lists) for dialup (first time only), the Easy IP router requests a single registered IP address from the access server at the central site via PPP/IPCP. (See Figure 2.)

Figure 2 Cisco Easy IP Router Requests a Dynamic Global IP Address



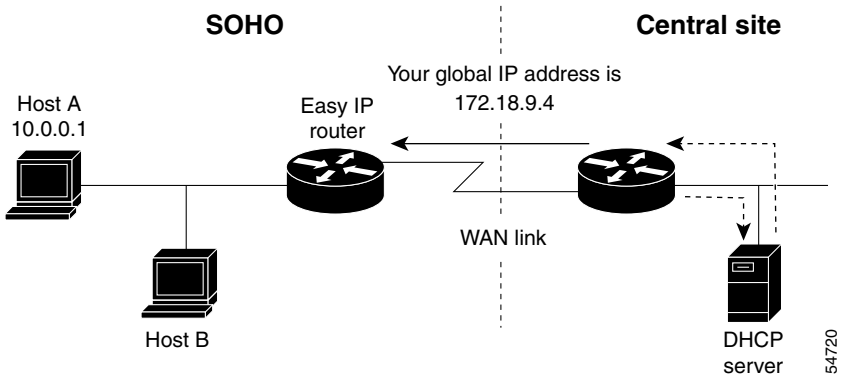
Step 2 The central site router replies with a dynamic global address from a local DHCP IP address pool. (See Figure 3.)

Figure 3 Dynamic Global IP Address Delivered to the Cisco Easy IP Router



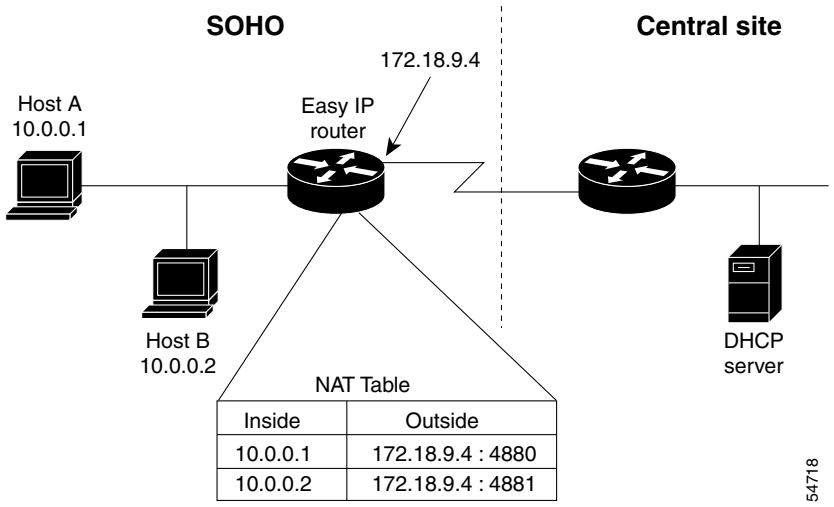
Step 3 The Cisco Easy IP router uses port-level NAT functionality to automatically create a translation that associates the registered IP address of the WAN interface with the private IP address of the client. (See Figure 4.)

Figure 4 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 5.)

Figure 5 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
```

Configuring Virtual Profiles

First Published: December 15, 1997
Last Updated: November 20, 2014

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.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Configuring Virtual Profiles" section on page 20.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Prerequisites for Configuring Virtual Profiles

Cisco recommends that unnumbered addresses be used in virtual template interfaces to ensure that duplicate network addresses are not created on virtual access interfaces (VAIs).

Restrictions for Configuring Virtual Profiles

The **virtual-profile** command was removed from Cisco IOS Release 12.2(34)SB and 12.2(33)XNE, because Cisco 10000 series routers do not support the full VAIs these releases create and configuration errors could occur.

Information About Configuring Virtual Profiles

This section provides information about virtual profiles for use with virtual access interfaces and how virtual profiles work. Virtual profiles run on all Cisco IOS platforms that support Multilink PPP (MLP). Virtual profiles interoperate with Cisco dial-on-demand routing (DDR), MLP, and dialers such as ISDN. To configure virtual profiles, you should understand the following concepts:

Virtual Profiles Overview

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.

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 1, 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 1, “(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 1 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, page 5—Applies the virtual template.
- Case 2: Virtual Profiles Configured by AAA, page 6—Applies the user-specific interface configuration received from the AAA server.
- Case 3: Virtual Profiles Configured by Virtual Template and AAA Configuration, page 6—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, page 7—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.

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.

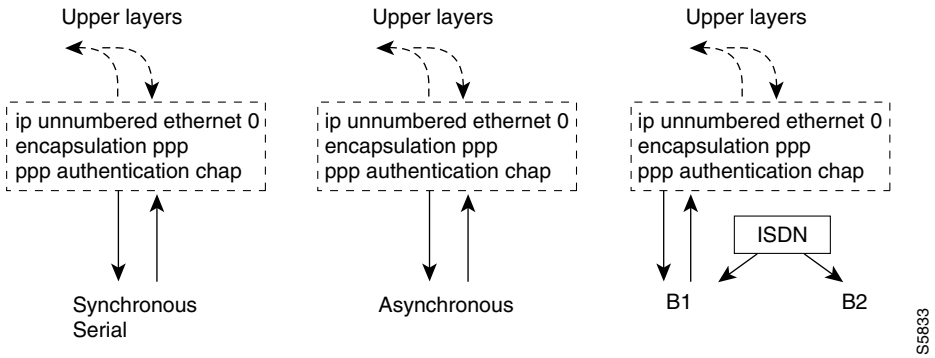
Then the router interprets the lines in the AAA authorization approval response from the server as Cisco IOS commands to apply to the virtual profile for the user.

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 1, 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 1 Virtual Profiles by Virtual Template



See the "Configuring Virtual Profiles by Virtual Template" section on page 8 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.

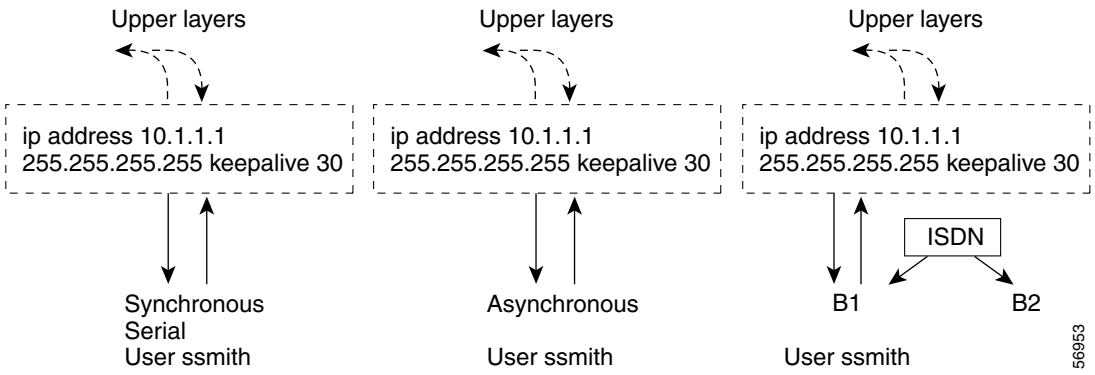
Suppose, for example, that the router interpreted the response by the AAA server as including only the following two commands for this user:

```
ip address 10.10.10.10 255.255.255.255
keepalive 30
```

In Figure 2, 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 2 Virtual Profiles by AAA Configuration



See the "Configuring Virtual Profiles by AAA Configuration" section on page 9 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.

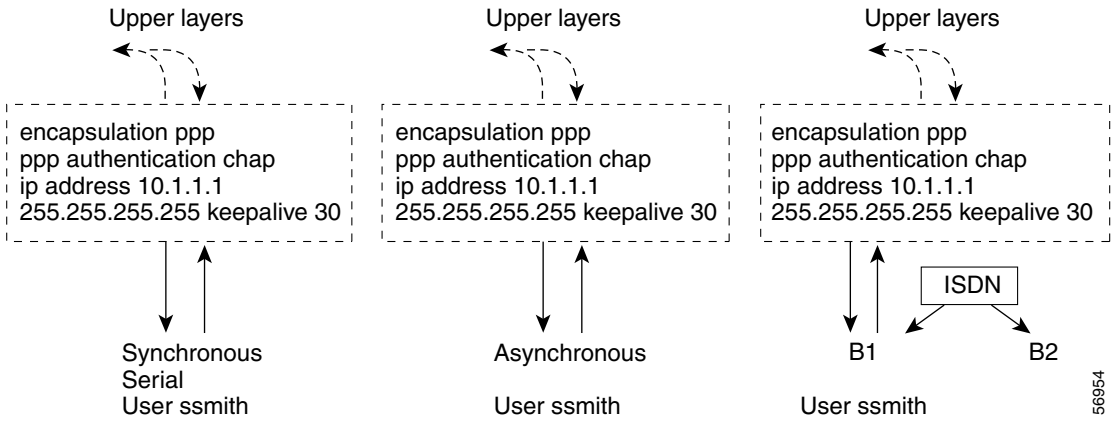
The router performs the following tasks in order:

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 3 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 3 Virtual Profiles by Both Virtual Template and AAA Configuration



See the "Configuring Virtual Profiles by Both Virtual Template and AAA Configuration" section on page 9 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, page 8 (as required)
- Configuring Virtual Profiles by AAA Configuration, page 9 (as required)
- Configuring Virtual Profiles by Both Virtual Template and AAA Configuration, page 9 (as required)
- Troubleshooting Virtual Profile Configurations, page 11 (as required)

Note Do not define a DDR dialer profile for a user if you intend to define virtual profiles for the user.

Configuring Virtual Profiles by Virtual Template

To configure virtual profiles by virtual template, complete these two tasks:

- Creating and Configuring a Virtual Template Interface, page 8
- Specifying a Virtual Template Interface for Virtual Profiles, page 9

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:

	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:

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*.
- 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:

Command	Purpose
`Router(config)# virtual-profile aaa` **Note** Effective with Cisco IOS Release 12.2(34)SB and 12.2(33)XNE, the **virtual-profile aaa** command is not available in Cisco IOS software. In releases later than Cisco IOS Release 12.2, the router automatically creates virtual profiles when AAA attributes require a profile.	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:

	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:

	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 Configuring per-User Configuration feature.

Troubleshooting Virtual Profile Configurations

To troubleshoot the virtual profiles configurations, use any of the following **debug** commands:

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 cloning`	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: Example, page 11
- Virtual Profiles Configured by AAA Configuration: Example, page 13
- Virtual Profiles Configured by Virtual Templates and AAA Configuration: Example, page 14
- Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway: Example, page 15

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 (User1 and User2) occur in all these examples. Note the different configuration allowed to them in each of the four examples.

User1 is a normal user and can dial in to BRI 0 only. User2 is a privileged user who can dial in to BRI 0 and BRI 1. If User2 dials into BRI 1, the dialer profile will be used. If User2 dials into BRI 0, virtual profiles will be used. Because User1 does not have a dialer profile, only virtual profiles can be applied to User1.

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: Example

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 User1 and User2.
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 User1 1111
 dialer map ip 10.1.1.3 name User2 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 1 for DDR to dial out to User2.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name User2
 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: Example

The following example shows the router configuration for virtual profiles by AAA and the AAA server configuration for user-specific interface configurations. User1 and User2 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 User1 and User2

```
User1 Password = "welcome"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 75\nip address 192.16.100.100
 255.255.255.0",
User2 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 User1 and User2.
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 User1 1111
 dialer map ip 10.1.1.3 name User2 2222
 dialer-group 1
```

```
 ppp authentication chap
!
! Configure dialer interface 1 for DDR to dial out to User2.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name User2
 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: Example

The following example shows how virtual profiles can be configured by both virtual templates and AAA configuration. User1 and User2 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 User1 and User2

```
User1 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"
User2 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 User1 and User2.
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 User1 1111
 dialer map ip 10.1.1.3 name User2 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR to dial out to User2.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name User2
 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: Example

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 User1 and User2, the two users. Specifically, User1 and User2 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 User1 and User2

```
User1 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",
User2 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 User1 and User2.
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 User1 1111
 dialer map ip 10.1.1.3 name User2 2222
 dialer-group 1
 ppp authentication chap
!
!Configure dialer interface 1 for DDR to dial out to User2.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name User2
 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
```

Additional References

The following sections provide references related to configuring virtual profiles.

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Commands List, All Releases*
Dial commands	*Cisco IOS Dial Command Reference*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Configuring Virtual Profiles

Table 2 lists the features in this module and provides links to specific configuration information.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 2 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 2 Feature Information for Configuring Virtual Profiles

Feature Name	Releases	Feature Information
Configuring Virtual Profiles	11.3	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 Interface Template Service

First Published: May 10, 2001
Last Updated: November 20, 2014

The Virtual Interface Template Service feature provides a generic service that can be used to apply predefined interface configurations (virtual interface template services) in creating and freeing virtual access interfaces dynamically, as needed.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Virtual Interface Template" section on page 10.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Restrictions for Virtual Interface Template Service

The following restrictions apply for configuring the virtual interface template service feature:

- Although a system can generally support many virtual interface template services, 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.
- You cannot reuse virtual interface templates. You need to create different templates for different interface configurations.
- You cannot directly configure virtual access interfaces. You need to configure a virtual access interface by configuring a virtual interface template service or including the configuration information of the user 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 you; 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 interface template service.

For a complete description of the virtual interface service commands mentioned in this chapter, refer to the *Cisco IOS Dial Technologies Command Reference*. For information about all Cisco IOS commands, use the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or a Cisco IOS master commands list.

Information About Virtual Interface Template Service

To configure the virtual interface template service, you should understand the following concepts:

- Virtual Interface Template Service Overview, page 2
- Benefits of Virtual Interface Template Service, page 3
- Features that Use Virtual Interface Template Service, page 3
- Selective Virtual Access Interface Creation, page 4

Virtual Interface Template Service Overview

Virtual interface template services 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.

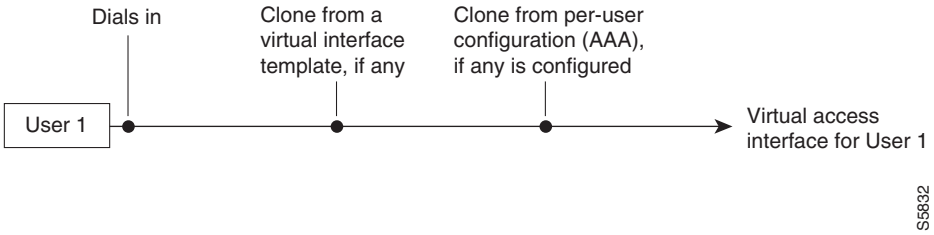
A virtual interface template service is a logical entity—a configuration for a serial interface but not tied to a physical interface—that can be applied dynamically as needed. Virtual access interfaces are virtual interfaces that are created, configured dynamically (for example, by *cloning* a virtual interface template service), used, and then freed when no longer needed.

Virtual interface template services 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; The result of using template and AAA configuration sources is a virtual access interface uniquely configured for a specific dial-in user.

Figure 1 illustrates that a device can create a virtual access interface by first using the information from a virtual interface template service (if any is defined for the application) and then using the information in a per-user configuration.

Figure 1 Possible Configuration Sources for Virtual Access Interfaces



Benefits of Virtual Interface Template Service

The virtual interface template service is intended primarily for customers with large numbers of dial-in users and provides the following benefits:

- Easy maintenance: It allows customized configurations to be predefined and then applied dynamically when the specific need arises.
- Scalability: It 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.
- Consistency and configuration ease: It allows the same predefined template to be used for all users dialing in for a specific application.
- Efficient device operation: It frees the virtual access interface memory for another dial-in use when the call from the user ends.

Features that Use Virtual Interface Template Service

The following features use virtual interface template service to create virtual access interfaces dynamically:

- Virtual Private Dialup Networks (VPDNs)
- Virtual interface templates for protocol translation
- PPP over ATM

Virtual interface templates are supported on all platforms that support these features.

To create and configure a virtual interface template interface, compete the tasks in the "Creating and Configuring a Virtual Interface Template" section on page 4. To apply a virtual interface template service, refer to the specific feature that applies the virtual interface template.

All prerequisites depend on the feature that is applying a virtual interface template to create a virtual access interface. Virtual interface template services themselves have no other prerequisites.

Selective Virtual Access Interface Creation

You can configure a device 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 for RADIUS or TACACS+ can be processed without a virtual access interface being created by a device.

To determine whether a virtual access interface is created, ensure the following exists:

- AAA per-user configuration
- Support for link interface support direct per-user AAA

A virtual access interface is created if 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 is not created if the following conditions are not satisfied:

- There is no AAA per-user configuration.
- There is AAA per-user configuration and the link interface does support direct per-user AAA (such as asynchronous).

How to Configure a Virtual Interface Template

This section contains the following tasks:

- Creating and Configuring a Virtual Interface Template, page 4 (required)
- Monitoring and Maintaining a Virtual Access Interface, page 5 (required)

Note The order in which you create virtual interface template service and configure the features that use the templates and profiles is not important. They must exist, however, before someone calling in can use them.

Creating and Configuring a Virtual Interface Template

To create and configure a virtual interface template service, use the **interface virtual-template** command.

Note Configuring the **ip address** command within a virtual interface template service is not recommended. Configuring a specific IP address in a virtual interface template can result in the establishment of erroneous routes and the loss of IP packets.

Other PPP configuration commands can be added to the virtual interface 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 interface template interfaces, except the **shutdown** and **dialer** commands.

For virtual interface template examples, see the "Configuration Examples for Virtual Interface Template" section on page 6 section.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **ip unnumbered ethernet** *number*
5. **encapsulation ppp**
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual-template** *number* **Example:** `Device(config)# interface virtual-template 0/0`	Creates a virtual interface template and enters interface configuration mode.
Step 4	**ip unnumbered ethernet** number **Example:** `Device(config-if)# ip unnumbered ethernet 0/0`	Enables IP without assigning a specific IP address on the LAN.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual interface template.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Monitoring and Maintaining a Virtual Access Interface

When a virtual interface template 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 **show interfaces virtual-access** and c**lear interface virtual-access** commands.

SUMMARY STEPS

1. **enable**
2. **show interfaces virtual-access** *number*
3. **clear interface virtual-access** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show interfaces virtual-access** *number* **Example:** `Device# show interfaces virtual-access 3`	Displays the configuration of the virtual access interface.
Step 3	**clear interface virtual-access** *number* **Example:** `Device# clear interface virtual-access 3`	Tears down the virtual access interface and frees the memory for other dial-in uses.

Configuration Examples for Virtual Interface Template

The following sections provide virtual interface template configuration examples:

Virtual Interface Template: Example

The following example shows how to verify a virtual interface template configuration.

Note Effective with Cisco Release 12.4(11)T, the **l2f protocol** command was removed in Cisco IOS software.

```
Device# show interfaces virtual-access 1

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: Example

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-template 1
 ip unnumbered Ethernet 0
 no ip directed-broadcast
 no keepalive
 ppp authentication chap
 ppp multilink
```

Selective Virtual Access Interface Configuration for RADIUS per User: Example

This example shows how to create AAA per-user configuration for a RADIUS user profile. When a AAA per-user configuration for a RADIUS user profile exists, a virtual access interface is configured automatically.

```
RADIUS user profile:
        name1 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"
```

Selective Virtual Access Interface Configuration for TACACS+ per User: Example

This example shows how to create AAA per-user configuration for a TACACS+ user profile:

```
user = name1 {
                name = "name1"
                global = cleartext test
                service = PPP protocol= ip {
                        inacl#1="deny 10.10.10.10 0.0.0.0"
                        inacl#1="permit any"
                }
        }
```

Additional References

The following sections provide references related to the Virtual Interface Template Service feature.

Related Documents

Related Topic	Document Title
Dial interfaces, controllers and lines	"Overview of Dial Interfaces, Controllers, and Lines" module in the *Cisco IOS Dial Technologies Configuration Guide*
Dial commands	*Cisco IOS Dial Technologies Command Reference*

Standards

Standard	Title
None	—

MIBs

MIB	MIBs Link
None	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	—

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Virtual Interface Template

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Virtual Interface Templates

Feature Name	Releases	Feature Information
Virtual Interface Template Service	11.2(1) 12.2(14)S 12.2(27)SBA 12.2(33)SRE Cisco IOS XE 3S	Virtual interface template service can be configured independently of any physical interface and applied dynamically to create virtual access interfaces. The following sections provide information about this feature: • Information About Virtual Interface Template Service, page 2 • Creating and Configuring a Virtual Interface Template, page 4 • Monitoring and Maintaining a Virtual Access Interface, page 5 The following commands were introduced or modified: **clear interfaces virtual-access**, **interface virtual-template**, and **show interfaces virtual-access**

Configuring Media-Independent PPP and Multilink PPP

First Published: May 10, 2001
Last Updated: November 20, 2014

The Configuring Media-Independent PPP and Multilink PPP module describes how to configure PPP and Multilink PPP (MLP) features on any interface. This module also describes address pooling for point-to-point links, which is available on all asynchronous serial, synchronous serial, and ISDN interfaces.

Multilink PPP provides a method for spreading traffic across multiple physical WAN links.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "Feature Information for Configuring Media-Independent PPP and Multilink PPP" section on page 50.

Use Cisco Feature Navigator to find information about platform support and Cisco IOS and Catalyst OS software image support. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Prerequisites for Media-Independent PPP and Multilink PPP

Understanding PPP and multilink operations.

Information About Media-Independent PPP and Multilink PPP

To configure the Media-Independent PPP and Multilink PPP, you should understand the following concepts:

Point-to-Point Protocol

Point-to-Point Protocol (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

The implementation of PPP supports authentication using Challenge Handshake Authentication Protocol (CHAP), Microsoft Challenge Handshake Authentication Protocol (MS-CHAP), or Password Authentication Protocol (PAP), and the option 4, and option 5, and Magic Number configuration options.

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.

CHAP or PPP 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 was 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 hostname 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 hostname 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 hostname 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. The username and password specified in the authentication request are accepted, and 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:

- Enable PPP encapsulation.
- Enable CHAP or PAP on the interface.

For CHAP, configure hostname authentication and the secret password for each remote system with which authentication is required.

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. The 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.

The router sends an acknowledgment (ACK) confirming MPPC compression negotiation.

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.

The IP address pooling feature 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.

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 Serial Line Internet Protocol (SLIP) EXEC command—An asynchronous dialup user can enter a peer IP address or hostname 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.
- 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 authentication, authorization, and accounting (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:

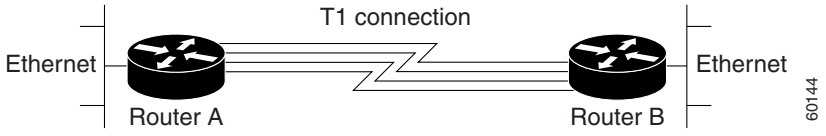
1. AAA/TACACS+ provided address or addresses from the pool named by AAA/TACACS+
2. An address from a local IP address pool or DHCP (typically not allocated unless no other address exists)
3. Dialer map lookup address (not done unless no other address exists)
4. Address from an EXEC-level PPP or SLIP 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)

MLP on Synchronous Serial Interfaces

Address pooling is available on all asynchronous serial, synchronous serial, ISDN BRI, and ISDN PRI interfaces that are running PPP and PPPoX sessions.

MLP provides characteristics are most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure 1 shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

Figure 1 Inverse Multiplexing Application Using Multilink PPP



How to Configure Media-Independent PPP and Multilink PPP

This section includes the following procedures:

Configuring PPP and MLP

Perform the following task in interface configuration mode to configure PPP on a serial interface (including ISDN). This task is required for 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, page 7
- Configuring Compression of PPP Data, page 9
- Configuring IP Address Pooling
- Disabling or Reenabling Peer Neighbor Routes
- Configuring Multilink PPP
- Configuring Multilink PPP
- Configuring MLP Interleaving
- Creating a Multilink Bundle
- Assigning an Interface to a Multilink Bundle
- Disabling PPP Multilink Fragmentation

See the "Monitoring and Maintaining PPP and MLP Interfaces" section on page 35 for tips on maintaining PPP. See the "Configuration Examples for PPP and MLP" section on page 36 to understand how to implement PPP and MLP in your network.

Enabling PPP Encapsulation

The **encapsulation ppp** command enables PPP on serial lines to encapsulate IP and other network protocol datagrams.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **configure fastethernet** *number*
4. **encapsulation ppp**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *number* **Example:** `Device(config)# interface fastethernet 0/0`	Enters interface configuration mode.
Step 4	**encapsulation ppp** **Example:** `Device(config-if) # encapsulation ppp`	Enables PPP encapsulation. **Note** PPP echo requests are used as keepalives to minimize disruptions to the end users of your network. Use the **no keepalive command** to disable echo requests.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode.

Enabling CHAP or PAP Authentication

To enable CHAP or PAP authentication, perform the steps mentioned in this section.

Caution If you use a list name that has not been configured with the **aaa authentication ppp** command, you disable PPP on the line.

For an example of CHAP, see the section ""Examples: CHAP with an Encrypted Password:" section on page 36". CHAP is specified in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*.

For information about MS-CHAP, see *MS-CHAP Support*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *number*
4. **ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**]
5. **ppp use-tacacs** [**single-line**]

   or

   aaa authentication ppp

6. **exit**
7. **username** *name* [**user-maxlinks** *link-number*] **password** *secret*
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *number* **Example:** `Device(config)# interface fastethernet 0/0`	Enters Interface Configuration mode.
Step 4	**ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**] **Example:** `Device(config-if)# ppp authentication chap`	Defines the authentication methods supported and the order in which they are used. **Note** Use the **ppp authentication chap** command only with TACACS or extended TACACS. **Note** With AAA configured on the router and list names defined for AAA, the *list-name* optional argument can be used with AAA/TACACS+. Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.

Step 5	**ppp use-tacacs** [**single-line**]
	or
	aaa authentication ppp
	
	Example:
	`Device(config-if)# ppp use-tacacs single-line`
	or
	`Device(config-if)# aaa authentication ppp`
		Configure TACACS on a specific interface as an alternative to global host authentication.

Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.

	Command or Action	Purpose
Step 7	**username** *name* [**user-maxlinks** *link-number*] **password** *secret* **Example:** `Device(config)# username name user-maxlinks 1 password password1`	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.
Step 8	**end** **Example:** `Device(config)# end`	Exits global configuration mode and enters privileged EXEC mode.

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.

To configure compression of PPP data, perform the steps in this section.

Software Compression

Software compression is available on 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface fastethernet** *number*
4. **encapsulation PPP**
5. **compress** [**predictor | stac | mppc** [**ignore-pfc**]]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface fastethernet** *number* **Example:** `Device(config)# interface fastethernet 0/0`	Enters interface configuration mode.
Step 4	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.
Step 5	**compress** [**predictor** | **stac** | **mppc** [**ignore-pfc**]] **Example:** `Device(config-if)# compress predictor`	Enables compression.
Step 6	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode.

Configuring Microsoft Point-to-Point Compression

Perform this task to configure MPCC. This will help you set MPPC once PPP encapsulation is configured on the router.

Prerequisites

Ensure that PPP encapsulation is enabled before you configure MPPC. For information on how to configure PPP encapsulation, see the "Enabling PPP Encapsulation" section on page 6".

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).

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface serial** *number*
4. **compress** [**mppc** [**ignore-pfc**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface serial** *number* **Example:** `Device(config)# interface serial 2/0`	Enters interface configuration mode.
Step 4	**compress** [**mppc** [**ignore-pfc**]] **Example:** `Device(config-if)# compress mppc`	Enables encapsulation of a single protocol on the serial line. • The **ignore-pfc** keyword instructs the router to ignore the protocol field compression flag negotiated by Link Control Protocol (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.

Examples

Following is 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

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

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 the following sections:

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:

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.

Perform this task to enable DHCP as the global default mechanism.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip address-pool dhcp-proxy-client**
4. **ip dhcp-server** [*ip-address* | *name*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip address-pool dhcp-proxy-client** **Example:** `Device(config)# ip address-pool dhcp-proxy-client`	Specifies the DHCP client-proxy feature as the global default mechanism. • The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses. **Note** 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. The DHCP servers provide temporary IP addresses.
Step 4	**ip dhcp-server** [*ip-address* | *name*] **Example:** `Device(config)# ip dhcp-server 209.165.201.1`	(Optional) Specifies the IP address of a DHCP server for the proxy client to use.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode.

Defining Local Address Pooling as the Global Default Mechanism

Perform this task to define local address pooling as the global default mechanism.

Note 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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip address-pool local**
4. **ip local pool** {*named-address-pool* | **default**} *first-IP-address* [*last-IP-address*] [**group** *group-name*] [**cache-size** *size*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip address-pool local** **Example:** `Device(config)# ip address-pool local`	Specifies local address pooling as the global default mechanism.
Step 4	**ip local pool** {*named-address-pool* | d**efault**} *first-IP-address* [*last-IP-address*] [**group** *group-name*] [**cache-size** *size*] **Example:** `Device(config)# ip local pool default 192.0.2.1`	Creates one or more local IP address pools.

Controlling DHCP Network Discovery

Perform the steps in this section to allow peer routers to dynamically discover Domain Name System (DNS) and NetBIOS name server information configured on a DHCP server using PPP IPCP extensions.

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 timeout period from 3 to 15 seconds, or leave the default timeout period at 15 seconds. The default for the **informs** and **discovers** keywords is 0, which disables the transmission of these messages.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip dhcp-client network-discovery informs** *number-of-messages* **discovers** *number-of-messages* **period** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip dhcp-client network-discovery informs** *number-of-messages* **discovers** *number-of-messages* **period** *seconds* **Example:** `Device(config)# ip dhcp-client network-discovery informs 2 discovers 2 period 2`	Provides control of the DHCP network discovery mechanism by allowing the number of DHCP Inform and Discover messages to be sent, and a timeout period for retransmission, to be configured.

Configuring IP Address Assignment

Perform this task to configure IP address allignment.

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.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip local pool** {**named-address-pool** | **default**} {*first-IP-address* [*last-IP-address]*} [**group** *group-name*] [**cache-size** *size*]}
4. **interface** *type number*
5. **peer default ip address pool** *pool-name-list*
6. **peer default ip address pool dhcp**
7. **peer default ip address** *ip-address*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip local pool** {*named-address-pool* | d**efault**} {*first-IP-address* [*last-IP-address*]} [**group** *group-name*] [**cache-size** *size*]} **Example:** `Device(config)# ip local pool default 192.0.2.0`	Creates one or more local IP address pools.
Step 4	**interface** *type number* **Example:** `Device(config)# interface ethernet 2/0`	Specifies the interface and enters interface configuration mode.
Step 5	**peer default ip address pool** *pool-name-list* **Example:** `Device(config-if)# peer default ip address pool 2`	Specifies the pool or pools for the interface to use.
Step 6	**peer default ip address pool dhcp** **Example:** `Device(config-if)# peer default ip address pool dhcp`	Specifies DHCP as the IP address mechanism on this interface.
Step 7	**peer default ip address** *ip-address* **Example:** `Device(config-if)# peer default ip address 192.0.2.2`	Specifies the IP address to assign to all dial-in peers on an 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 Link Access Procedure, Balanced (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, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **no peer neighbor-route**
5. **peer neighbor-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface ethernet 0/1`	Specifies the interface and enters interface configuration mode.
Step 4	**no peer neighbor-route** **Example:** `Device(config-if)# no peer neighbor-route`	Disables creation of neighbor routes.
Step 5	**peer neighbor-route** **Example:** `Device(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 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, page 18
- Configuring MLP on Asynchronous Interfaces, page 19
- Configuring MLP on a Single ISDN BRI Interface, page 21
- Configuring MLP on Multiple ISDN BRI Interfaces, page 23
- Configuring MLP Using Multilink Group Interfaces, page 25
- Changing the Default Endpoint Discriminator, page 27

Configuring MLP on Synchronous Interfaces

To configure Multilink PPP on synchronous interfaces, you configure the synchronous interfaces to support PPP encapsulation and Multilink PPP.

Perform this task to configure a synchronous interface.

SUMMARY STEPS

1. **enable**
2. **configuration terminal**
3. **interface serial** 1
4. **no ip address**
5. **encapuslation ppp**
6. **ppp multilink**
7. **pulse-time** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface serial** *number* **Example:** `Device(config)# interface serial 1`	Specifies an asynchronous interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Specifies no IP address for the interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 7	**pulse-time** *seconds* **Example:** `Device(config-if)# pulse-time 60`	Enables pulsing data terminal ready (DTR) signal intervals on an interface. **Note** Repeat these steps for additional synchronous interfaces, as needed.

Configuring MLP on Asynchronous Interfaces

Perform the following steps in this section to configure an asynchronous interface to support DDR and PPP encapsulation and then configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

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.

Note To configure a dialer interface to support PPP encapsulation and Multilink PPP, use the **dialer load-threshold** command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface async** *number*
4. **no ip address**
5. **dialer in-band**
6. **dialer rotary-group** *number*
7. **dialer load-threshold** *load* [**inbound** | **outbound** | **either**]

8. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface async** *number* **Example:** `Device(config)# interface async 0/0`	Specifies an asynchronous interface and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Specifies no IP address for the interface.
Step 5	`Device(config-if)#` **encapsulation ppp** **Example:** `Device# configure terminal`	Enables PPP encapsulation.
Step 6	**dialer in-band** **Example:** `Device(config-if)# encapsulation ppp`	Enables DDR on the interface.
Step 7	**dialer rotary-group** *number* **Example:** `Device(config-if)# dialer rotary-group 1`	Includes the interface in a specific dialer rotary group.
Step 8	**dialer load-threshold** *load* [**inbound** | **outbound** | **either**] **Example:** `Device(config-if)# dialer load-threshold 100`	Configures bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.
Step 9	**ppp multilink** **Example:** `Device(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.

Perform this task to enable PPP on an ISDN BRI interface.

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 "Example: MLP on One ISDN BRI Interface" section on page 43.

Note When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a high idle timer. The **dialer-load threshold 1** command does not keep a multilink bundle of *n* links connected indefinitely, and the **dialer-load threshold 2** command does not keep a multilink bundle of two links connected indefinitely.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface bri** *number*
4. **ip address** *ip-address mask* [**secondary**]
5. **encapsulation ppp**
6. **dialer idle-timeout** *seconds* [**inbound** | **either**]
7. **dialer load-threshold** *load*
8. **dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
9. **dialer-group** *group-number*
10. **ppp authentication pap**
11. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface bri** *number* **Example:** `Device(config)# interface bri 1`	Specifies an interface and benters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 192.0.2.0 255.255.255.224`	Provides an appropriate protocol address for the interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**dialer idle-timeout** *seconds* [**inbound** | **either**] **Example:** `Device(config-if)# dialer idle-timeout 60`	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 7	**dialer load-threshold** *load* **Example:** `Device(config-if)# dialer load-threshold 60`	Specifies the dialer load threshold for bringing up additional WAN links.
Step 8	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]] **Example:** `Device(config-if)# dialer map protocol 192.0.2.1`	Configures the ISDN interface to call the remote site.
Step 9	**dialer-group** *group-number* **Example:** `Device(config-if)# dialer-group 3`	Controls access to this interface by adding it to a dialer access group.
Step 10	**ppp authentication pap** **Example:** `Device(config-if)# ppp authentication pap`	(Optional) Enables PPP authentication.
Step 11	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Configures MLP on the dialer rotary group.

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, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *dialer number*
4. **ip address** *address mask*
5. **encapsulation ppp**
6. **dialer in-band**
7. **dialer idle-timeout** *seconds* [**inbound** | **either**]
8. **dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]
9. **dialer rotary-group** *number*
10. **dialer load-threshold** *load*
11. **dialer-group** *number*
12. **ppp authentication chap**
13. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface dialer** *number* **Example:** `Device(config)# interface dialer 1`	Specifies the dialer rotary interface and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.0.2.0 255.255.255.224`	Specifies the protocol address for the dialer rotary interface.

	Command or Action	Purpose
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**dialer in-band** **Example:** `Device(config-if)# dialer in-band`	Specifies in-band dialing.
Step 7	**dialer idle-timeout** *seconds* [**inbound** | **either**] **Example:** `Device(config-if)# dialer idle-timeout 60`	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 8	**dialer map** *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]] **Example:** `Device(config-if)# dialer map protocol 192.0.2.1`	Maps the next hop protocol address and name to the dial string needed to reach it.
Step 9	**dialer rotary-group** *number* **Example:** `Device(config-if)# dialer rotary-group 1`	Adds the interface to the rotary group.
Step 10	**dialer load-threshold** *load* **Example:** `Device(config-if)# dialer load-threshold 2`	Specifies the dialer load threshold, using the same threshold as the individual BRI interfaces.
Step 11	**dialer-group** *number* **Example:** `Device(config-if)# dialer-group 2`	Controls access to the interface by adding it to a dialer access group.
Step 12	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables PPP CHAP authentication.
Step 13	**ppp multilink** **Example:** `Device(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.

Repeat Steps 1 through 9 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 does not keep a multilink bundle of *n* links connected indefinitely and the **dialer load-threshold 2** command does not keep a multilink bundle of two links connected indefinitely.)

Note Prior to Cisco IOS Release 12.1, 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 module "Configuring Peer-to-Peer DDR with Dialer Profiles" in this module, 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 "Example: MLP on Multiple ISDN BRI Interfaces" section on page 43.

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.

Note If a multilink group interface has one member link, the amount of bandwidth available will not change when a multilink interface is shut down. Therefore, you can shut down the multilink interface by removing its link.

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.

Perform the following tasks in this section to configure the multilink group. For an example of how to configure MLP over an ATM PVC using a multilink group, see the section "Example: MLP Using Multilink Group Interfaces over ATM" section on page 44.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*

4. **ip address** *address mask*
5. **encapsulation ppp**
6. **exit**
7. **interface virtual template** *number*
8. **ppp multilink group** *group-number*
9. **exit**
10. **interface atm** *interface-number.subinterface-number* **point-to-point**
11. **pvc** *vpi/vli*
12. **protocol ppp virtual-template** *name*
13. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 2`	Creates a multilink bundle and enters interface configuration mode to configure the bundle.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.0.2.1 255.255.255.224`	Sets a primary IP address for an interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 7	**interface virtual template** *number* **Example:** `Device(config)# interface virtual template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.

	Command or Action	Purpose
Step 8	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink group 2`	Restricts a physical link to joining only a designated multilink group interface.
Step 9	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 10	**interface atm** *interface-number.subinterface-number* **point-to-point** **Example:** `Device(config)# interface atm 1.2 point-to-point`	Configures an ATM interface and enters interface configuration mode.
Step 11	**pvc** *vpi/vci* **Example:** `Device(config-if)# pvc 1/100`	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.
Step 12	**protocol ppp virtual-template** *name* **Example:** `Device(config-if-atm-vc)# protocol ppp virtual-template 2`	Configures VC multiplexed encapsulation on a PVC.
Step 13	**end** **Example:** `Device(config-if-atm-vc)# end`	Exits ATM virtual circuit configuration mode.

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 hostname (or stack group name, if this interface is a Stack Group Bidding Protocol, or SGBP, group member).

Perform this task to override or change the default endpoint discriminator. For an example of how to change the default endpoint discriminator, see the "Example: Changing the Default Endpoint Discriminator" section on page 37.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virutal template** *number*

4. **ppp multilink endpoint** {**hostname** | **ip** *ipaddress* | **mac** *LAN-interface* | **none** | **phone** *telephone-number* | **string char-string**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface virtual template** *number* **Example:** `Device(config)# interface virtual template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces and enters interface configuration mode.
Step 4	**ppp multilink endpoint** {**hostname** | **ip** *ipaddress* | **mac** *LAN-interface* | **none** | **phone** *telephone-number* | **string** *char-string*} **Example:** `Device(config-if)# ppp multilink endpoint ip 192.0.2.0`	Overrides or changes the default endpoint discriminator the system uses when negotiating the use of MLP with the peer.

Configuring MLP Interleaving

Perform the following tasks to configure MLP and interleaving on a configured and operational interface or virtual interface template.

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 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.
- Configure MLP and interleaving on the interface or template.

Note Fair queueing, which is enabled by default, must remain enabled on the interface.

Note 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)
```

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual template** *number*
4. **ppp multilink**
5. **ppp multilink interleave**
6. **ppp multilink fragment delay** *milliseconds*
7. **ip rtp reserve** *lowest-udp-port range-of-ports* [*maximum-bandwidth*]
8. **exit**
9. **multilink virtual-template** *virtual-template-number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface virtual template** *number* **Example:** `Device(config)# interface virtual template 1`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.
Step 4	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 5	**ppp multilink interleave** **Example:** `Device(config-if)# configure terminal`	Enables interleaving of packets among the fragments of larger packets on an MLP bundle.
Step 6	**ppp multilink fragment delay** *milliseconds* **Example:** `Device(config-if)# ppp multilink fragment delay 50`	Specifies a maximum size, in units of time, for packet fragments on an MLP bundle.
Step 7	**ip rtp reserve** *lowest-udp-port range-of-ports* [*maximum-bandwidth*] **Example:** `Device(config-if)# ip rtp reserve 1 2`	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 8	**exit** **Example:** `Device(config-if)# exit`	Exits interface configuration mode.
Step 9	**multilink virtual-template** *virtual-template-number* **Example:** `Device(config)# multilink virtual-template 1`	For virtual templates only, applies the virtual template to the multilink bundle. **Note** This step is not used for ISDN or dialer interfaces.

Configuring MLP Inverse Multiplexer and Distributed MLP

The distributed MLP (dMLP) feature combines T1/E1 lines in a WAN line card on a Cisco 7600 series router into a bundle that has the combined bandwidth of the multiple T1/E1 lines. 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 is not supported on the Cisco 7600 series router. 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 dMLP can bundle multiple T1 lines in a CT3 or CE3 interface or channelized STM1. Bundling is more economical than purchasing an inverse multiplexer, and eliminates the need to configure another piece of equipment.

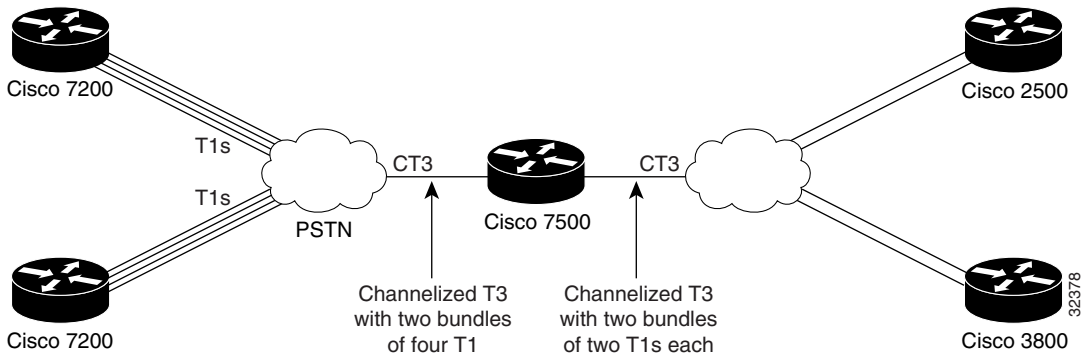
This feature supports the CT3 CE3 data rates without taxing the Route Processor (RP) and CPU by moving the data path to the line card. This feature also allows remote sites to purchase multiple T1 lines instead of a T3 line, which is especially useful when the remote site does not need the bandwidth of an entire T3 line.

This feature allows multilink fragmentation to be disabled, so multilink packets are sent using Cisco Express Forwarding on all platforms, if fragmentation is disabled. Cisco Express Forwarding is supported with fragmentation enabled or disabled.

Note If a router cannot send out all the packets (some packets are dropped by Quality of Service (QoS)), late drops occur. These late drops are displayed when the **show interface** command is executed.
If there is no service policy on the dMLP interface, when a **ppp multilink interleave** is configured on the dMLPPP interface, a QoS policy is enabled internally.

Figure 2 shows a typical network using a dMLP link. The Cisco 7600 series router is connected to the network with a CT3 line that has been configured with dMLPP 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 2 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 Cisco Express Forwarding 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 following restrictions apply to the dMLP feature:

Note Distributed MLP is supported only for member links configured at T1/E1 or subrate T1/E1 speeds. Channelized STM-1/T3/T1 interfaces also support dMLP at T1/E1 or subrate T1/E1 speeds. Distributed MLP is not supported for member links configured at clear-channel T3/E3 or higher interface speeds.

- T1 and E1 lines cannot be mixed in a bundle.
- T1 lines in a bundle should have the same bandwidth.
- All lines in a bundle must reside on the same port adapter.
- MLP bundles across FlexWAN or Enhanced FlexWAN port adapters are not supported.
- Hardware compression is not supported.
- Encryption is not supported.
- Software compression is not recommended because CPU usage would void performance gains.
- The maximum differential delay supported is 50 milliseconds (ms).
- Fragmentation is not supported on the transmit side.
- dMLP across shared port adapters (SPAs) is not supported.
- Hardware and software compression is not supported.
- Encryption is not supported.
- The maximum differential delay supported is 50 ms when supported in hardware, and 100 ms when supported in software.

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:

- Creating a Multilink Bundle, page 32 (required)
- Assigning an Interface to a Multilink Bundle, page 33 (required)
- Disabling PPP Multilink Fragmentation, page 35 (optional)

Creating a Multilink Bundle

Perform the following tasks to create a multilink bundle.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address mask*
5. **encapsulation ppp**
6. **ppp multilink**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**ip address** *address mask* **Example:** `Device(config-if)# ip address 192.0.2.9 255.255.255.224`	Assigns an IP address to the multilink interface.
Step 5	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.

Assigning an Interface to a Multilink Bundle

Perform this task to assign an interface to a multilink bundle.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group number*
4. **no ip address**
5. **keepalive**
6. **encapsulation ppp**
7. **ppp multilink group** *group-number*
8. **ppp multilink**
9. **ppp authentication chap**
10. **pulse-time seconds**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.
Step 4	**no ip address** **Example:** `Device(config-if)# no ip address`	Removes any specified IP address.
Step 5	**keepalive** **Example:** `Device(config-if)# keepalive`	Sets the frequency of keepalive packets.
Step 6	**encapsulation ppp** **Example:** `Device(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 7	**ppp multilink group** *group-number* **Example:** `Device(config-if)# ppp multilink 12`	Restricts a physical link to joining only the designated multilink-group interface.
Step 8	**ppp multilink** **Example:** `Device(config-if)# ppp multilink`	Enables Multilink PPP.
Step 9	**ppp authentication chap** **Example:** `Device(config-if)# ppp authentication chap`	(Optional) Enables CHAP authentication.
Step 10	**pulse-time** *seconds* **Example:** `Device(config-if)# pulse-time 10`	(Optional) Configures DTR signal pulsing.

Caution Do not install a router to the peer address while configuring an MLP lease line. This installation can be disabled when **no ppp peer-neighbor-route** command is used under the MLPPP bundle interface.

Disabling PPP Multilink Fragmentation

Perform the following task to disable PPP multilink fragmentation.

SUMMARY STEPS

1. **enable**
2. **configuration terminal**
3. **interface multilink** *group number*
4. **ppp multilink fragment disable**
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface multilink** *group-number* **Example:** `Device(config)# interface multilink 10`	Assigns a multilink group number and enters interface configuration mode.

Step 4	**ppp multilink fragment disable**
	
	Example:
	`Device(config-if)#`
	`ppp multilink fragment disable`
		(Optional) Disables PPP multilink fragmentation.

Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits privileged EXEC mode.

Monitoring and Maintaining PPP and MLP Interfaces

Perform this task to display MLP and MMP bundle information.

SUMMARY STEPS

1. **enable**
2. **show ppp multilink**
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ppp multilink** **Example:** `Device# show ppp multilink`	Displays MLP and MMP bundle information.
Step 3	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode.

Configuration Examples for PPP and MLP

The following sections provide various PPP configuration examples:

Examples: CHAP with an Encrypted Password:

The following examples show how to enable CHAP on serial interface 0 of three devices:

Configuration of Device yyy

```
hostname yyy
interface serial 0
```

```
 encapsulation ppp
 ppp authentication chap
username xxx password secretxy
username zzz password secretzy
```

Configuration of Device xxx

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password secretxy
username zzz password secretxz
```

Configuration of Device 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 are encrypted and the display looks similar to the following:

```
hostname xxx
interface serial 0
 encapsulation ppp
 ppp authentication chap
username yyy password 7 121F0A18
username zzz password 7 1329A055
```

Example: Changing the Default Endpoint Discriminator

The following partial example changes the MLP endpoint discriminator from the default CHAP hostname C-host1 to the E.164-compliant telephone number 555-0100:

```
.
.
.
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 555-0100
.
.
.
```

Example: DHCP Network Control

The following partial example shows how to add the **ip dhcp-client network-discovery** command to the "Example: IP Address Pooling" section on page 38 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
.
.
.
```

Example: IP Address Pooling

The following example shows how to configure 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 Password
!
scheduler interval 100
end
```

Example: MPPC Interface Configuration

The following example shows how to configure 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
```

Examples: MLP

This section contains the following MLP examples:

- Example: MLP on Synchronous Serial Interfaces, page 41
- Example: MLP on One ISDN BRI Interface, page 43
- Example: MLP on Multiple ISDN BRI Interfaces, page 43
- Example: MLP Inverse Multiplexer Configuration, page 44
- Example: MLP Using Multilink Group Interfaces over ATM, page 44
- Example: Changing the Default Endpoint Discriminator, page 37

Example: MLP on Synchronous Serial Interfaces

The following example shows how the configuration commands are used to create the inverse multiplexing application:

Device A Configuration

```
hostname DeviceA
!
!
username DeviceB 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
```

Device B Configuration

```
hostname DeviceB
!
!
username DeviceB 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
```

Example: MLP on One ISDN BRI Interface

The following example shows how to enable MLP on BRI interface 0. When a BRI is configured, no dialer rotary group configuration is required, because an ISDN interface is a rotary group by default.

```
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 user1 81012345678901
 dialer-group 1
 ppp authentication pap
 ppp multilink
```

Example: MLP on Multiple ISDN BRI Interfaces

The following example shows how to configure 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 BRI 0
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI 1
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI 2
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface Dialer 0
 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 user1 broadcast 81012345678901
 dialer load-threshold 30 either
 dialer-group 1
 ppp authentication chap
 ppp multilink
```

Example: MLP Using Multilink Group Interfaces over ATM

The following example shows how to configure 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
 exit
 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
```

Example: MLP Inverse Multiplexer Configuration

This example shows how to verify the display information of the newly created multilink bundle:

```
Device# 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
```

Example: MLP Interleaving and Queueing for Real-Time Traffic

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 name1 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
```

Example: Multilink Interface Configuration for Distributed MLP

In the following example, four multilink interfaces are created with distributed Cisco Express Forwarding 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
```

Example: PAP commands for a one way authentication

The following example shows how to authenticate PAP commands for a one way authentication scenario:

Note Only the relevant sections of the configuration are shown.

```
Calling Side (Client) Configuration
interface BRI0

! --- BRI interface for the dialout.

 ip address negotiated
 encapsulation ppp

! --- Use PPP encapsulation. This command is a required for PAP.

 dialer string 3785555 class 56k

! --- Number to dial for the outgoing connection.

 dialer-group 1
 isdn switch-type basic-ni
 isdn spid1 51299611110101 9961111
 isdn spid2 51299622220101 9962222
 ppp authentication pap callin

! --- Use PAP authentication for incoming calls.
! --- The callin keyword has made this a one-way authentication scenario.
! --- This router (client) will not request that the peer (server) authenticate
! --- itself back to the client.
```

```
 ppp pap sent-username PAPUSER password 7 <deleted>

! --- Permit outbound authentication of this router (client) to the peer.
! --- Send a PAP AUTH-REQ packet to the peer with the username PAPUSER and password.
! --- The peer must have the username PAPUSER and password configured on it.

Receiving Side (Server) Configuration
username PAPUSER password 0 cisco

! --- Username PAPUSER is the same as the one sent by the client.
! --- Upon receiving the AUTH-REQ packet from the client, we will verify that the
! --- username and password match the one configured here.

interface Serial0:23

! --- This is the D-channel for the PRI on the access server receiving the call.

 ip unnumbered Ethernet0
 no ip directed-broadcast
 encapsulation ppp

! --- Use PPP encapsulation. This command is a required for PAP.

 dialer-group 1
 isdn switch-type primary-ni
 isdn incoming-voice modem
 peer default ip address pool default
 fair-queue 64 256 0
 ppp authentication pap

! --- Use PAP authentication for incoming calls.
! --- This router (server) will request that the peer authenticate itself to us.
! --- Note: the callin option is not used as this router is not initiating the call.
```

Example: T3 Controller Configuration for an MLP Multilink Inverse Multiplexer

The following example shows how to configure the T3 controller and create four channelized interfaces:

```
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
```

Example: User Maximum Links Configuration

The following example shows how to configure the username user1 and establish a maximum of five connections. user1 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 user1 can connect to.

```
aaa new-model
aaa authorization network default local
enable secret password1
```

```
enable password password2
!
username user1 user-maxlinks 5 password password3
!
interface Serial0/0:23
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap
 ppp multilink
!
interface Serial1/0
 ip address 209.165.201.1 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer map ip 10.2.2.13 name user1 12345
 dialer-group 1
 ppp authentication chap
!
interface Dialer0
 ip address 209.165.200.225 255.255.255.0
 encapsulation ppp
 dialer remote-name user1
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp authentication chap
 ppp multilink
!
dialer-list 1 protocol ip permit
```

Additional References

The following sections provide references related to the Configuring Media-Independent PPP and Multilink PPP feature.

Related Documents

Related Topic	Document Title
Asynchronous SLIP and PPP	*"Configuring Asynchronous SLIP and PPP"* module in the *Cisco IOS Dial Technologies Configuration Guide*
MCHAP	*MS-CHAP Support*
Dial commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Dial Technologies Command Reference.*

RFCs

RFC	Title
RFC 1994	PPP Challenge Handshake Authentication Protocol (CHAP)

Technical Assistance

Description	Link
The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies. To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.	http://www.cisco.com/techsupport

Feature Information for Configuring Media-Independent PPP and Multilink PPP

Table 1 lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn. An account on Cisco.com is not required.

Note Table 1 lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

Table 1 Feature Information for Configuring Media-Independent PPP and Multilink PPP

Feature Name	Releases	Feature Information
Multilink PPP	11.2(1) 12.2(8)T 11.2(6)P 12.1(3)T 12.3(13)BC 12.2(27)SBB 12.2(31)SB2 15.0(1)M 12.2(33)SRE 15.2(2)S Cisco IOS XE Release 3.14S	Multilink PPP provides a method for spreading traffic across multiple physical WAN links. The following sections provide information about this feature: • Information About Media-Independent PPP and Multilink PPP, page 2 • How to Configure Media-Independent PPP and Multilink PPP, page 6 The following commands were introduced or modified: **ppp multilink**, **ppp multilink group**.

Customer Profile Idle Timer Enhancements for Interesting Traffic

Feature History

Release	Modification
12.2(4)T	This feature was introduced.
12.2(11)T	This feature was implemented on Cisco access server platforms.

This document describes the Asynchronous Line Monitoring feature feature. It includes the following sections:

Feature Overview

Before Cisco IOS Release 12.2(4)T, only the dialer idle timer could be reset for *interesting* traffic on a dialer interface. The Asynchronous Line Monitoring feature feature available in Cisco IOS Release 12.2(4)T supports a PPP idle timer based on interesting traffic for dialer interfaces. (Existing PPP idle timer behavior is not changed when traffic is not classified.) New commands and functionality provided with this feature also address idle timer issues for virtual access dialup network (VPDN) sessions, which use virtual access (projected) interfaces and rely on the PPP idle timer mechanism.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

The Resource Pool Manager (RPM) per-customer profile dialer idle timer function works with Multilink PPP (MLP) and Multichassis Multilink PPP (MMP), providing that the master bundle interface is not a virtual access (projected) interface. For virtual access interfaces such as those used in a VPDN or with MMP where the dialer idle timer cannot be used, you can now classify the IP traffic that resets the PPP idle timer. A named access list is also supported.

Additionally, because RPM customer profiles are applied on a per-Dialed Number Identification Service (DNIS) basis and allow for configuring a per-customer profile dialer idle timer, the Asynchronous Line Monitoring feature feature associates idle timers based on call type and DNIS.

The idle timer implementation in the Asynchronous Line Monitoring feature feature specifies that for calls terminated on a network access server, a virtual access interface is cloned from the virtual template. This virtual access interface is linked to a physical interface on which is running a dialer timer. If the PPP idle timer is configured on the virtual template or provided by an authentication, authorization, and accounting (AAA) per-user interface configuration, the result is two idle timers, as follows:

- A PPP idle timer on the virtual access interface.
- A dialer idle timer on the physical interface.

Neither the dialer idle timer nor the PPP idle timer will run when the idle timer in the per-user configuration is set to 0. When the per-user idle timer is set to some value besides 0, that value overrides all local idle timer configurations.

Benefits

The Asynchronous Line Monitoring feature feature provides the following system idle timer benefits:

- Resets the PPP idle timer based on interesting inbound or outbound IP traffic for virtual access interfaces on Layer 2 Tunnel Protocol (L2TP) access concentrators (LACs) and L2TP network servers (LNSs).
- Associates the dialer timer with interesting traffic within RPM customer profiles.
- Applies the user idle-timer value RADIUS attribute 28 across all interfaces associated with the call.

Restrictions

The PPP idle timer can classify IP traffic only.

Supported Platforms

See the next section for information about Feature Navigator and how to use this tool to determine the platforms and software images in which this feature is available.

Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

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

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated when major Cisco IOS software releases and technology releases occur. As of May 2001, Feature Navigator supports M, T, E, S, and ST releases. You can access Feature Navigator at the following URL:

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

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

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 the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Configuration Tasks

See the following sections for configuration tasks for the Asynchronous Line Monitoring feature feature. Each task in the list is identified as either required or optional:

- Configuring an RPM Template to Accept Dialer Interface Timers (required)
- Configuring a PPP Idle Timer Based on Interesting IP Traffic (required)
- Configuring the Idle Timer in a RADIUS Profile (optional)
- Verifying the Asynchronous Line Monitoring feature (optional)

Configuring an RPM Template to Accept Dialer Interface Timers

To configure a template to accept dialer interface timers, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# template name`	Accesses the template configuration mode for configuring a particular customer profile template.
Step 2	`Router(config-template)# dialer idle-timeout seconds`	Sets the dialer idle timeout period in a virtual template interface.
Step 3	`Router(config-template)# dialer-group dialer-list-number`	Controls access by configuring an interface to belong to a specific dialing group.

Configuring a PPP Idle Timer Based on Interesting IP Traffic

To configure a PPP idle timer based on *interesting* IP traffic, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.
Step 2	`Router(config-if)# ppp timeout idle time`	Sets PPP idle timeout parameters on the virtual template interface.
Step 3	`Router(config-if)# ip idle-group {access-list-number | access-list-name} {in | out}`	Configures interesting inbound traffic (using the **in** keyword) or outbound traffic (using the **out** keyword) on a virtual template interface for the PPP idle timer.

See the configurations included in the "Configuration Examples" section for additional commands that you might configure.

Configuring the Idle Timer in a RADIUS Profile

To set the idle timer from AAA, configure the following RADIUS profile:

```
aaaa-idle Password = "password"
 Service-Type = Framed,
 Framed-Protocol = PPP,
 Idle-timeout = 60
```

Verifying the Asynchronous Line Monitoring feature

To verify that the Asynchronous Line Monitoring feature is configured correctly, perform the following verification steps:

Step 1 To display the idle time configured, and any remote caller that is connected and its IP address, enter the **show caller timeou**t EXEC command:

```
Router# show caller timeout

    Line       User       Host(s)              Idle       Location
*  0 con 0                 idle                 00:00:00
  47 tty 47    st-5300-c3 Async interface      00:00:15   PPP: 11.1.1.2

  Interface      User        Mode                       Idle     Peer Address
```

Step 2 Enter the **show caller timeou**t EXEC command again. Notice that the **show caller timeout** command displays the idle timeout configured as 20 seconds:

```
Router# show caller timeout
                                  Session   Idle      Disconnect
  Line         User               Timeout   Timeout   User in
  con 0        -                  -         -         -
  tty 47       st-5300-c3         -         00:30:00  00:29:43
  As47         st-5300-c3         -         00:00:20  now
```

Step 3 Continue entering the **show caller timeout** command. The displays show the timers counting down and then disconnecting.

```
Router# show caller timeout
                                  Session   Idle      Disconnect
  Line         User               Timeout   Timeout   User in
  con 0        -                  -         -         -
  tty 47       st-5300-c3         -         00:30:00  00:29:43
  As47         st-5300-c3         -         00:00:20  now
Router#

Router# show caller timeout
                                  Session   Idle      Disconnect
  Line         User               Timeout   Timeout   User in
  con 0        -                  -         -         -
  tty 47       -                  -         00:30:00  00:29:41

Router# show caller timeout
                                  Session   Idle      Disconnect
  Line         User               Timeout   Timeout   User in
  con 0        -                  -         -         -
  tty 47       -                  -         00:30:00  00:29:38

Router# show caller timeout
                                  Session   Idle      Disconnect
  Line         User               Timeout   Timeout   User in
  con 0        -                  -         -         -
```

Troubleshooting Tips

To troubleshoot the Asynchronous Line Monitoring feature feature, use the following debugging commands:

- **debug cca**
- **debug aaa authen**
- **debug aaa author**
- **debug aaa per-user**

- **debug ppp authen**
- **debug ppp neg**
- **debug radius**
- **debug isdn q931**
- **debug dialer detail**
- **debug vaccess**
- **debug vprofile**

Monitoring and Maintaining the Asynchronous Line Monitoring feature

To monitor and maintain the Asynchronous Line Monitoring feature feature, use the following EXEC commands:

Command	Purpose
`Router# show caller`	Displays caller information.
`Router# show ip access-list`	Displays the contents of all current IP access lists.
`Router# show users`	Displays information about the active lines on the router.

Configuration Examples

This section provides the following configuration examples:

- Two Templates with Different Dialer Idle Timer Settings Example
- Resetting the Dialer Idle Timer with Interesting Traffic Example
- Network Access Server Extended Configuration Example

Two Templates with Different Dialer Idle Timer Settings Example

The following partial example shows how to configure two customer profiles, each with different templates. Notice that each template sets the dialer idle timer differently:

```
resource-pool enable
!
resource-pool profile customer prf_cust_1
 limit base-size all
 limit overflow-size 0
 dnis group dnis_g1
 source template template1
!
resource-pool profile customer prf_cust_2
 limit base-size all
 limit overflow-size 0
 dnis group dnis_g2
 source template template2
!
```

```
template template1
 dialer idle-timeout 45
 dialer-group 1
!
template template2
 dialer idle-timeout 90
 dialer-group 2
!
dialer dnis group dnis_g1
 number 11111111
!
dialer dnis group dnis_g2
 number 22222222
```

Resetting the Dialer Idle Timer with Interesting Traffic Example

The following partial example shows how to configure an RPM customer profile that sets the dialer idle timer in a virtual template interface based on either inbound or outbound traffic:

```
resource-pool enable
!
resource-pool profile customer prf_cust_1
 limit base-size all
 limit overflow-size 0
 dnis group dnis_g1
 source template template1
!
template template1
 dialer idle-timeout 45 either
 dialer-group 1
!
dialer dnis group dnis_g1
 number 1231231234
```

Network Access Server Extended Configuration Example

The following example shows the configuration for a Cisco AS5300 series access server, which is part of a large-scale dial-out configuration. Notice that on virtual template interface 1 the PPP idle timer is configured to reset only on interesting inbound traffic, and that both dialer interface idle timers are set to 60 seconds:

```
hostname 5300
!
aaa new-model
aaa authentication ppp default local group radius none
aaa authorization network default local group radius none
!
username 4500 password 0 cisco
username 5300 password 0 cisco
username 2500-1 password 0 cisco
username 2500-2 password 0 cisco
username LAC password 0 cisco
username LNS password 0 cisco
username SGBP password 0 cisco
spe 1/0 2/9
 firmware location system:/ucode/mica_port_firmware
!
resource-pool enable
!
```

```
resource-pool group resource modem
 range port 1/0 1/48
!
resource-pool group resource data
 range limit 20
!
resource-pool profile customer cust
 limit base-size all
 limit overflow-size 0
 resource modem speech
 resource data digital
 dnis group dnis_g7
 source template1
!
dialer dnis group dnis_g7
 number 11111112

ip subnet-zero
!
sgbp group MMP
sgbp member 2500-2 10.0.38.3
sgbp ppp-forward
!
vpdn enable
no vpdn logging
!
isdn switch-type primary-5ess
!
template template1
 dialer idle-timeout 27
 dialer-group 1
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback1
 ip address 192.168.14.1 255.255.255.255
 no ip route-cache
 no ip mroute-cache
!
interface Ethernet0
 ip address 10.0.38.14 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface Virtual-Template1
 ip unnumbered Loopback1
 peer default ip address pool local_pool
 ppp authentication chap callin
 ppp chap hostname name
 ppp timeout idle 60
 ip idle-group 101 in
 ip idle-group 102 in
 ppp multilink
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
 ip mroute-cache
 load-interval 30
 dialer load-threshold 1 outbound
```

```
 dialer-group 2
 isdn switch-type primary-5ess
 isdn incoming-voice modem
 ppp authentication chap callin
 ppp multilink
!
interface Async1
 ip unnumbered Loopback1
 encapsulation ppp
 dialer in-band
 dialer rotary-group 1
 dialer-group 1
 async mode dedicated
!
interface Dialer1
 ip unnumbered Loopback1
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 60
 dialer-group 1
 peer default ip address pool local_pool
 ppp authentication chap callin
 ppp chap hostname name
!
ip local pool local_pool 10.1.14.1 10.1.14.254
ip classless
ip route 172.0.0.0 255.0.0.0 Ethernet0
ip route 192.168.0.0 255.255.255.0 10.0.38.1
no ip http server
!
access-list 101 deny   icmp any any
access-list 101 permit ip any any
!
access-list 102 deny   tcp any any
access-list 102 permit ip any any
!
dialer-list 1 protocol ip list 101
dialer-list 2 protocol ip list 102
dialer-list 3 protocol ip permit
!
access-list 101 permit icmp any any
access-list 102 deny   ip any any

radius-server host 172.69.70.72 auth-port 1645 acct-port 1646
radius-server retransmit 3
radius-server key cisco
!
line con 0
 exec-timeout 0 0
 transport input none
line 1 2
 no exec
 exec-timeout 0 0
 autoselect ppp
 script dialer dial
 script reset reset
 modem InOut
 modem autoconfigure discovery
 transport input all
line 2 240
 no exec
 exec-timeout 0 0
 transport input all
line aux 0
```

```
line vty 0 4
!
end
```

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Dial Technologies Command Reference* at http://www.cisco.com/en/US/docs/ios/dial/command/reference/dia_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

New Commands

- **ip idle-group**
- **dialer-group (template)**
- **dialer idle-timeout (template)**
- **ppp timeout idle (template)**

Modified Command

- **dialer-list protocol**

Glossary

interesting packets—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.

Troubleshooting Enhancements for Multilink PPP over ATM Link Fragmentation and Interleaving

The Troubleshooting Enhancements for Multilink PPP over ATM Link Fragmentation and Interleaving enhance the output of the **show atm pvc**, **show multilink ppp**, and **show interfaces virtual-access** commands to display multilink PPP (MLP) over ATM link fragmentation and interleaving (LFI) information. This feature also introduces the **debug atm lfi** command, which can be used to display MLP over ATM LFI debugging information.

Feature History for Troubleshooting Enhancements for Multilink PPP over ATM LFI

Release	Modification
12.3(7)T	This feature was introduced.

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at http://www.cisco.com/go/fn. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

How to Troubleshoot Multilink PPP over ATM LFI

This section contains the following procedure:

Troubleshooting Multilink PPP over ATM LFI

Perform this task to display information about multilink PPP over ATM LFI connections.

Prerequisites

This task assumes that you have configured multilink PPP over ATM LFI in your network. For information about how to configure multilink PPP over ATM LFI, see the "Additional References" section on page 3.

SUMMARY STEPS

1. **enable**
2. **show atm pvc** *vpi/vci*
3. **show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]]
4. **show interfaces virtual-access** [*type number*]
5. **debug atm lfi**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show atm pvc** *vpi/vci* **Example:** `Router# show atm pvc 15/200`	Displays traffic, management, and MLP over ATM LFI information for the specified PVC.
Step 3	**show ppp multilink** [**active** | **inactive** | **interface** *bundle-interface* | [**username** *name*] [**endpoint** *endpoint*]] **Example:** `Router# show ppp multilink username blue`	Displays bundle information for MLP bundles.

	Command or Action	Purpose
Step 4	**show interfaces virtual-access** *number* **Example:** `Router# show interfaces virtual-access 3`	Displays status, traffic data, and configuration information about a specified virtual access interface. • Display will indicate if the interface is a member of a multilink PPP bundle.
Step 5	**debug atm lfi** **Example:** `Router# debug atm lfi`	Displays MLP over ATM LFI debug information.

Examples

See the **show atm pvc**, **show ppp multilink**, **show interfaces virtual-access**, and **debug atm lfi** command pages for examples of output and descriptions of the fields in the output. For information about where to find the command pages for these commands, see Command Reference, page 5.

Additional References

The following sections provide references related to multilink PPP over ATM LFI.

Related Documents

Related Topic	Document Title
LFI for multilink PPP configuration tasks	*"Configuring Link Fragmentation and Interleaving for Multilink PPP" chapter in the Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2*
LFI for ATM virtual circuits configuration tasks	*"Configuring Link Fragmentation and Interleaving for Frame Relay and ATM Virtual Circuits" chapter in the Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2*
Multilink PPP over ATM LFI commands	*Cisco IOS Quality of Service Solutions Command Reference*, Release 12.3 T
Multilink PPP configuration tasks	"PPP Configuration" section in the *Cisco IOS Dial Technologies Configuration Guide*, Release 12.3
Multilink PPP commands	*Cisco IOS Dial Technologies Command Reference*, Release 12.3 T
ATM configuration tasks	*"WAN Protocols" section in the Cisco IOS Wide-Area Networking Configuration Guide, Release 12.3*
ATM commands	*Cisco IOS Wide-Area Networking Command Reference*, Release 12.3 T

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

The following commands are introduced or modified in the feature or features documented in this module. For information about these commands, see the *Cisco IOS Dial Technologies Command Reference* at http://www.cisco.com/en/US/docs/ios/dial/command/reference/dia_book.html. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

- **debug atm lfi**
- **show atm pvc**
- **show interfaces virtual-access**
- **show ppp multilink**

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.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

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. To debug PPP connections using MS Callback, see the **debug ppp cbcp** command in the *Cisco IOS Debug Command Reference* publication.

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.

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 atlanta class dial1 81012345678901
 dialer-group 1
 ppp callback accept
 ppp authentication chap
!
map-class dialer dial1
 dialer callback-server username
```

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 dallas 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:

- ISDN Caller ID Callback Overview
- How to Configure ISDN Caller ID Callback
- Monitoring and Troubleshooting ISDN Caller ID Callback
- Configuration Examples for ISDN Caller ID Callback

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

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 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 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
```

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.
`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.

Command	Purpose
`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
```

Large-Scale Dial-Out (LSDO) VRF Aware

Feature History

Release	Modification
12.2(8)T	This feature was introduced.

This document describes the LSDO VRF Aware feature in Cisco IOS Release 12.2(8)T and includes the following sections:

Feature Overview

Currently, the Cisco large-scale dial-out (LSDO) feature is not supported in a Multiprotocol Label Switching (MPLS) virtual private network (VPN), which means it does not support tunneling protocols and cannot take advantage of cost benefits inherent in an MPLS VPN. (See the sections "Benefits" and the "Related Documents" for more details on the benefits of MPLS VPN.) Beginning with Cisco IOS Release 12.2(8)T, large-scale dial-out will support the Layer 2 Tunnel Protocol (L2TP) in an MPLS VPN.

The basic operation of large-scale dial-out relies on per-user static routes stored in an authentication, authorization, and accounting (AAA) server, and redistributed static and redistributed connected routes to put better routes pointing to the same remote network or host on the alternate network access server (NAS).

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

A static route is manually configured on a NAS. If the static route that pointed to the next hop of the NAS has a name, that name with the -out suffix attached becomes the profile name.

When a packet arrives on a dialer interface where a static map is not configured, the dial string is retrieved from the AAA server. The query made to the AAA server is based on the destination IP address of the packet received.

When using L2TP VPN large-scale dial-out, overlapping IP addresses are often present in virtual routing and forwarding instances (VRFs), so that a unique key is needed to retrieve the correct route from the AAA server. With VPDN as a dial-out resource, a virtual access interface is created for maintaining each PPP session. Software prior to Cisco IOS Release 12.2(8)T did not update the VRF information on the virtual access interface; rather, this information was cloned from the dialer interface.

In the Cisco IOS Release 12.2(8)T software, the VRF table identifier is retrieved from the incoming packet and is mapped to the VRF name. This VRF name and the destination IP address are combined to make the unique key needed to retrieve the dial string and other user profile information from the AAA server. When response from the AAA server is received and the virtual access interface is created, the virtual access interface is updated with VRF information that was retrieved from the incoming packet. As with profile names on dialer interfaces, the IP address and VRF name combination with the -out suffix attached becomes the profile name for large-scale dial-out in MPLS VPN using L2TP.

Note Another way to build a unique key is to use the name of the IP route. In this situation, the key is made from the IP route name and VRF name combination with the -out suffix attached. Refer to the technical note listed in the "Related Documents" section for more information.

Benefits

Layer 2 Tunneling Technologies Trim Costs by Forwarding Calls over the Internet

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 Public Switched Telephone Network (PSTN) by using the Internet. Instead of connecting directly to the network by using the PSTN, access VPN users need only use the PSTN to connect to the Internet service provider (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 cost savings for the customer.

The MPLS VPN Model Simplifies Network Routing Configuration

The MPLS VPN model simplifies network routing by allowing VPN services to be supported in service provider networks. An MPLS VPN user can generally employ the backbone of the service provider as the default route in communicating with all of the other VPN sites.

The customer outsources the responsibility for the information technology (IT) infrastructure to an ISP that maintains the pool of modems the remote users dial in to, the access servers, and the internetworking expertise. The customer is responsible only for authenticating its users and maintaining its network.

L2TP Large-Scale Dial-Out Benefits from MPLS VPN Environment

The unique key created from the VRF name and the destination IP address allows retrieval of the dial string and other user profile information from a AAA server using L2TP in an MPLS VPN environment.

Restrictions

Cisco IOS Release 12.2(8)T supports *only* L2TP large-scale dial-out, and this feature makes it possible to retrieve *only* the dialer string that large-scale dial-out needs to construct the dynamic dialer map. This feature cannot create virtual access interfaces in the large-scale dial-out environment.

Related Documents

Additional information about configuring networks that can take advantage of this feature can be found in the following Cisco IOS documentation:

- *Cisco IOS Dial Technologies Command Reference*, Release 12.2.
- *Cisco IOS Dial Technologies Configuration Guide*, Release 12.2. Refer to the chapter "Configuring Large-Scale Dial-Out" in the part "Dial Access Specialized Features," and the chapter "Configuring Virtual Private Networks" in the part "Virtual Templates, Profiles, and Networks."
- *Cisco IOS Switching Services Command Reference*, Release 12.2.
- *Cisco IOS Switching Services Configuration Guide*, Release 12.2. Refer to the chapter "Multiprotocol Label Switching Overview" in the part "Multiprotocol Label Switching."

Supported Platforms

Use Cisco's Feature Navigator tool to determine which platforms support the Asynchronous Line Monitoring feature feature.

Determining Platform Support Through Feature Navigator

Cisco IOS software is packaged in feature sets that support specific platforms. To get updated information regarding platform support for this feature, access Feature Navigator. Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

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

To access Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions at http://www.cisco.com/register.

Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Feature Navigator home page at the following URL:

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

Supported Standards, MIBs, and RFCs

Standards

None

MIBs

None

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 the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

None

Prerequisites

No new Cisco IOS commands are introduced with the Asynchronous Line Monitoring feature feature. Before configuring this feature, read through the chapters listed in the "Related Documents" section, to be sure you know how to configure VPDNs, dialer interfaces, and MPLS, then use the examples in the section "Configuration Examples" to help you determine the configuration you need for your network.

Configuration Tasks

No new configuration tasks are required for configuring the Asynchronous Line Monitoring feature feature. See the sections "Prerequisites" and "Related Documents" for more information.

Monitoring and Maintaining Asynchronous Line Monitoring feature

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

To monitor and maintain Asynchronous Line Monitoring feature feature, use the following EXEC commands:

Command	Purpose
`Router# show dialer`	Displays general diagnostic information for interfaces configured for DDR.
`Router# show ip protocols vrf`	Displays the routing protocol information associated with a VRF.
`Router# show ip route vrf`	Displays the IP routing table associated with a VPN routing and VRF forwarding instance.
`Router# show ip vrf`	Displays the set of defined VRF instances and associated interfaces.
`Router# show vpdn`	Displays information about active L2F protocol tunnel and L2F message identifiers in a VPDN.
`Router# show vpdn domain`	Displays all VPDN domains and DNIS groups configured on the NAS.

Command	Purpose
`Router# show vpdn group`	Displays a summary of the relationships among VPDN groups and customer or VPDN profiles, and summarizes the configuration of a VPDN group including domain or DNIS, loadsharing information, and current session information.
`Router# show vpdn history failure`	Displays the content of the failure history table.
`Router# show vpdn multilink`	Displays the multilink sessions authorized for all VPDN groups.
`Router# show vpdn session`	Displays information about active L2TP or L2F sessions in a VPDN.
`Router# show vpdn tunnel`	Displays information about active L2TP or L2F tunnels in a VPDN.

Configuration Examples

This section contains partial sample configurations of the Asynchronous Line Monitoring feature feature. (Additional examples can be found in the technical note listed in the "Related Documents" section.)

In the following examples, VRF VPN_A has two hosts with the IP address 1.1.1.1 and 2.2.2.2 and, similarly, VRF VPN_B has two hosts with IP address 1.1.1.1 and 2.2.2.2. The AAA server is configured with a list containing "10.10.10.10-VPN_A-out" and "10.10.10.10-VPN_B-out" as keys to search on.

Note The network addresses used in the following configuration are examples only and will not work if tried in an actual network configuration.

LNS Configuration

This partial example configures L2TP dial-out tunnels to an L2TP access concentrator (LAC) from an L2TP network server (LNS):

```
request-dialout
   protocol l2tp
   rotary-group 1
! LAC IP address:
  initiate-to ip 172.16.0.2
  local name PE2_LNS
  l2tp tunnel password 7 13!9@61&
```

Dialer Configuration

This partial example configures the dialer interface:

```
interface Dialer 1
! Global IP address:
 ip address 10.10.10.10
 encapsulation ppp
 dialer in-band
 dialer aaa
 dialer vpdn
 dialer-group 1
 ppp authentication chap
```

Routing Configuration

This partial example configures the VRF static routes:

```
ip route vrf VPN_A 1.1.1.1 255.255.255.255 Dialer1
ip route vrf VPN_A 2.2.2.2 255.255.255.255 Dialer1
```

```
ip route vrf VPN_B 1.1.1.1 255.255.255.255 Dialer1
ip route vrf VPN_B 2.2.2.2 255.255.255.255 Dialer1
```

Command Reference

This feature uses no new or modified commands. For information about all Cisco IOS commands, go to the Command Lookup Tool at http://tools.cisco.com/Support/CLILookup or to the *Cisco IOS Master Commands List*.

Glossary

L2TP—Layer 2 Tunnel Protocol. A tunneling protocol that permits separating the remote access network function—terminating the PSTN circuit, for example—from the local network access operations such as authenticating and authorizing the remote user.

L2TP access concentrator—See LAC.

L2TP network server—See LNS.

LAC—L2TP access concentrator. A node that acts as one side of an L2TP tunnel endpoint and is a peer to the LNS. The LAC sits between an LNS and a remote system and forwards packets to and from each. Packets sent from the LAC to the LNS require tunneling with the L2TP protocol. The connection from the LAC to the remote system is either local or a PPP link.

Layer 2 Tunnel Protocol—See L2TP.

LNS—L2TP network server. A device that terminates an L2TP tunnel. It receives the remote user PPP connection over an L2TP tunnel. The LNS authenticates and authorizes the remote user and then forwards packets between the remote user and the data network.

MPLS—Multiprotocol Label Switching. Switching method that forwards IP traffic using a label. This label instructs the routers and the switches in the network where to forward the packets based on pre-established IP routing information.

Multiprotocol Label Switching—See MPLS.

NAS—network access server. A device that provides local network access to users across a remote access network such as the PSTN. For example, a NAS may provide access to a user dialing in from the PSTN to the data network, that is, it terminates the PSTN circuit, terminates the remote user PPP session, authenticates and authorizes the remote user, and finally forwards packets between the remote user and the data network.

network access server—See NAS.

virtual private dialup network—See VPDN.

virtual routing and forwarding instance—See VRF.

VPDN—virtual private dialup network. A type of access VPN that uses PPP to interface with the subscriber. VPDN enables the service provider to configure VPNs across an IP access network that connects to the VRFs on a PE. VPDN uses the Layer 2 Tunnel Protocol (L2TP) to extend or "tunnel" a PPP session across the IP access network.

VRF—virtual routing and forwarding instance. Identifies a separate VPN within a particular MPLS VPN network domain.