Cisco IOS Dial Technologies Configuration Guide

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

Cisco IOS Dial Technologies Configuration Guide

CONTENTS

About Cisco IOS and Cisco IOS XE Software Documentation

Last updated: August 6, 2008

This document describes the objectives, audience, conventions, and organization used in Cisco IOS and Cisco IOS XE software documentation, collectively referred to in this document as Cisco IOS documentation. Also included are resources for obtaining technical assistance, additional documentation, and other information from Cisco. This document is organized into the following sections:

Documentation Objectives

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

Audience

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

Documentation Conventions

In Cisco IOS documentation, the term *router* may be used to refer to various Cisco products; for example, routers, access servers, and switches. These and other networking devices that support Cisco IOS software are shown interchangeably in examples and are used only for illustrative purposes. An example that shows one product does not necessarily mean that other products are not supported.

This section includes the following topics:

Typographic Conventions

Cisco IOS documentation uses the following typographic conventions:

Convention	Description
^ or Ctrl	Both the ^ symbol and Ctrl represent the Control (Ctrl) key on a keyboard. For example, the key combination **^D** or **Ctrl-D** means that you hold down the Control key while you press the D key. (Keys are indicated in capital letters but are not case sensitive.)
string	A string is a nonquoted set of characters shown in italics. For example, when setting a Simple Network Management Protocol (SNMP) community string to *public*, do not use quotation marks around the string; otherwise, the string will include the quotation marks.

Command Syntax Conventions

Cisco IOS documentation uses the following command syntax conventions:

Convention	Description
bold	Bold text indicates commands and keywords that you enter as shown.
italic	Italic text indicates arguments for which you supply values.
[x]	Square brackets enclose an optional keyword or argument.
|	A vertical line, called a pipe, indicates a choice within a set of keywords or arguments.
[x | y]	Square brackets enclosing keywords or arguments separated by a pipe indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a pipe indicate a required choice.
[x {y | z}]	Braces and a pipe within square brackets indicate a required choice within an optional element.

Software Conventions

Cisco IOS uses the following program code conventions:

Convention	Description
`Courier font`	Courier font is used for information that is displayed on a PC or terminal screen.
`Bold Courier font`	Bold Courier font indicates text that the user must enter.
< >	Angle brackets enclose text that is not displayed, such as a password. Angle brackets also are used in contexts in which the italic font style is not supported; for example, ASCII text.
!	An exclamation point at the beginning of a line indicates that the text that follows is a comment, not a line of code. An exclamation point is also displayed by Cisco IOS software for certain processes.
[ ]	Square brackets enclose default responses to system prompts.

Reader Alert Conventions

The Cisco IOS documentation set uses the following conventions for reader alerts:

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

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

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

Documentation Organization

This section describes the Cisco IOS documentation set, how it is organized, and how to access it on Cisco.com. Included are lists of configuration guides, command references, and supplementary references and resources that make up the documentation set. The following topics are included:

Cisco IOS Documentation Set

Cisco IOS documentation consists of the following:

- Release notes and caveats provide information about platform, technology, and feature support for a release and describe severity 1 (catastrophic), severity 2 (severe), and severity 3 (moderate) defects in released Cisco IOS code. Review release notes before other documents to learn whether or not updates have been made to a feature.
- Sets of configuration guides and command references organized by technology and published for each standard Cisco IOS release.
  - Configuration guides—Compilations of documents that provide informational and task-oriented descriptions of Cisco IOS features.
  - Command references—Compilations of command pages that provide detailed information about the commands used in the Cisco IOS features and processes that make up the related configuration guides. For each technology, there is a single command reference that covers all Cisco IOS releases and that is updated at each standard release.
- Lists of all the commands in a specific release and all commands that are new, modified, removed, or replaced in the release.
- Command reference book for **debug** commands. Command pages are listed in alphabetical order.
- Reference book for system messages for all Cisco IOS releases.

Cisco IOS Documentation on Cisco.com

The following sections describe the documentation organization and how to access various document types.

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.

New Features List

The New Features List for each release provides a list of all features in the release with hyperlinks to the feature guides in which they are documented.

Feature Guides

Cisco IOS features are documented in feature guides. Feature guides describe one feature or a group of related features that are supported on many different software releases and platforms. Your Cisco IOS software release or platform may not support all the features documented in a feature guide. See the Feature Information table at the end of the feature guide for information about which features in that guide are supported in your software release.

Configuration Guides

Configuration guides are provided by technology and release and comprise a set of individual feature guides relevant to the release and technology.

Command References

Command reference books describe Cisco IOS commands that are supported in many different software releases and on many different platforms. The books are provided by technology. 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/all_release/all_mcl.html.

Cisco IOS Supplementary Documents and Resources

Supplementary documents and resources are listed in Table 2 on page xi.

Configuration Guides, Command References, and Supplementary Resources

Table 1 lists, in alphabetical order, Cisco IOS and Cisco IOS XE software configuration guides and command references, including brief descriptions of the contents of the documents. The Cisco IOS command references are comprehensive, meaning that they include commands for both Cisco IOS software and Cisco IOS XE software, for all releases. The configuration guides and command references support many different software releases and platforms. Your Cisco IOS software release or platform may not support all these technologies.

For additional information about configuring and operating specific networking devices, go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

Table 2 lists documents and resources that supplement the Cisco IOS software configuration guides and command references. These supplementary resources include release notes and caveats; master command lists; new, modified, removed, and replaced command lists; system messages; and the debug command reference.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS AppleTalk Configuration Guide *Cisco IOS XE AppleTalk Configuration Guide* *Cisco IOS AppleTalk Command Reference*	AppleTalk protocol.
Cisco IOS Asynchronous Transfer Mode Configuration Guide *Cisco IOS Asynchronous Transfer Mode Command Reference*	LAN ATM, multiprotocol over ATM (MPoA), and WAN ATM.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Bridging and IBM Networking Configuration Guide *Cisco IOS Bridging Command Reference* *Cisco IOS IBM Networking Command Reference*	• Transparent and source-route transparent (SRT) bridging, source-route bridging (SRB), Token Ring Inter-Switch Link (TRISL), and token ring route switch module (TRRSM). • Data-link switching plus (DLSw+), serial tunnel (STUN), block serial tunnel (BSTUN); logical link control, type 2 (LLC2), synchronous data link control (SDLC); IBM Network Media Translation, including Synchronous Data Logical Link Control (SDLLC) and qualified LLC (QLLC); downstream physical unit (DSPU), Systems Network Architecture (SNA) service point, SNA frame relay access, advanced peer-to-peer networking (APPN), native client interface architecture (NCIA) client/server topologies, and IBM Channel Attach.
Cisco IOS Broadband and DSL Configuration Guide *Cisco IOS XE Broadband and DSL Configuration Guide* *Cisco IOS Broadband and DSL Command Reference*	Point-to-Point Protocol (PPP) over ATM (PPPoA) and PPP over Ethernet (PPPoE).
Cisco IOS Carrier Ethernet Configuration Guide *Cisco IOS Carrier Ethernet Command Reference*	Connectivity fault management (CFM), Ethernet Local Management Interface (ELMI), IEEE 802.3ad link bundling, Link Layer Discovery Protocol (LLDP), media endpoint discovery (MED), and operations, administration, and maintenance (OAM).
Cisco IOS Configuration Fundamentals Configuration Guide *Cisco IOS XE Configuration Fundamentals Configuration Guide* *Cisco IOS Configuration Fundamentals Command Reference*	Autoinstall, Setup, Cisco IOS command-line interface (CLI), Cisco IOS file system (IFS), Cisco IOS web browser user interface (UI), basic file transfer services, and file management.
Cisco IOS DECnet Configuration Guide *Cisco IOS XE DECnet Configuration Guide* *Cisco IOS DECnet Command Reference*	DECnet protocol.
Cisco IOS Dial Technologies Configuration Guide *Cisco IOS XE Dial Technologies Configuration Guide* *Cisco IOS Dial Technologies Command Reference*	Asynchronous communications, dial backup, dialer technology, dial-in terminal services and AppleTalk remote access (ARA), large scale dialout, dial-on-demand routing, dialout, modem and resource pooling, ISDN, multilink PPP (MLP), PPP, virtual private dialup network (VPDN).
Cisco IOS Flexible NetFlow Configuration Guide *Cisco IOS Flexible NetFlow Command Reference*	Flexible NetFlow.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS H.323 Configuration Guide	Gatekeeper enhancements for managed voice services, Gatekeeper Transaction Message Protocol, gateway codec order preservation and shutdown control, H.323 dual tone multifrequency relay, H.323 version 2 enhancements, Network Address Translation (NAT) support of H.323 v2 Registration, Admission, and Status (RAS) protocol, tokenless call authorization, and VoIP gateway trunk and carrier-based routing.
Cisco IOS High Availability Configuration Guide *Cisco IOS XE High Availability Configuration Guide* *Cisco IOS High Availability Command Reference*	A variety of High Availability (HA) features and technologies that are available for different network segments (from enterprise access to service provider core) to facilitate creation of end-to-end highly available networks. Cisco IOS HA features and technologies can be categorized in three key areas: system-level resiliency, network-level resiliency, and embedded management for resiliency.
Cisco IOS Integrated Session Border Controller Command Reference	A VoIP-enabled device that is deployed at the edge of networks. An SBC is a toolkit of functions, such as signaling interworking, network hiding, security, and quality of service (QoS).
Cisco IOS Intelligent Service Gateway Configuration Guide *Cisco IOS Intelligent Service Gateway Command Reference*	Subscriber identification, service and policy determination, session creation, session policy enforcement, session life-cycle management, accounting for access and service usage, session state monitoring.
Cisco IOS Interface and Hardware Component Configuration Guide *Cisco IOS XE Interface and Hardware Component Configuration Guide* *Cisco IOS Interface and Hardware Component Command Reference*	LAN interfaces, logical interfaces, serial interfaces, virtual interfaces, and interface configuration.
Cisco IOS IP Addressing Services Configuration Guide *Cisco IOS XE Addressing Services Configuration Guide* *Cisco IOS IP Addressing Services Command Reference*	Address Resolution Protocol (ARP), Network Address Translation (NAT), Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Next Hop Address Resolution Protocol (NHRP).
Cisco IOS IP Application Services Configuration Guide *Cisco IOS XE IP Application Services Configuration Guide* *Cisco IOS IP Application Services Command Reference*	Enhanced Object Tracking (EOT), Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), IP Services, Server Load Balancing (SLB), Stream Control Transmission Protocol (SCTP), TCP, Web Cache Communication Protocol (WCCP), User Datagram Protocol (UDP), and Virtual Router Redundancy Protocol (VRRP).
Cisco IOS IP Mobility Configuration Guide *Cisco IOS IP Mobility Command Reference*	Mobile ad hoc networks (MANet) and Cisco mobile networks.
Cisco IOS IP Multicast Configuration Guide *Cisco IOS XE IP Multicast Configuration Guide* *Cisco IOS IP Multicast Command Reference*	Protocol Independent Multicast (PIM) sparse mode (PIM-SM), bidirectional PIM (bidir-PIM), Source Specific Multicast (SSM), Multicast Source Discovery Protocol (MSDP), Internet Group Management Protocol (IGMP), and Multicast VPN (MVPN).

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS IP Routing Protocols Configuration Guide *Cisco IOS XE IP Routing Protocols Configuration Guide* *Cisco IOS IP Routing Protocols Command Reference*	Border Gateway Protocol (BGP), multiprotocol BGP, multiprotocol BGP extensions for IP multicast, bidirectional forwarding detection (BFD), Enhanced Interior Gateway Routing Protocol (EIGRP), Interior Gateway Routing Protocol (IGRP), Intermediate System-to-Intermediate System (IS-IS), on-demand routing (ODR), Open Shortest Path First (OSPF), and Routing Information Protocol (RIP).
Cisco IOS IP SLAs Configuration Guide *Cisco IOS XE IP SLAs Configuration Guide* *Cisco IOS IP SLAs Command Reference*	Cisco IOS IP Service Level Agreements (IP SLAs).
Cisco IOS IP Switching Configuration Guide *Cisco IOS XE IP Switching Configuration Guide* *Cisco IOS IP Switching Command Reference*	Cisco Express Forwarding, fast switching, and Multicast Distributed Switching (MDS).
Cisco IOS IPv6 Configuration Guide *Cisco IOS XE IPv6 Configuration Guide* *Cisco IOS IPv6 Command Reference*	For IPv6 features, protocols, and technologies, go to the IPv6 "Start Here" document at the following URL: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-roadmap.html
Cisco IOS ISO CLNS Configuration Guide *Cisco IOS XE ISO CLNS Configuration Guide* *Cisco IOS ISO CLNS Command Reference*	ISO connectionless network service (CLNS).
Cisco IOS LAN Switching Configuration Guide *Cisco IOS XE LAN Switching Configuration Guide* *Cisco IOS LAN Switching Command Reference*	VLANs, Inter-Switch Link (ISL) encapsulation, IEEE 802.10 encapsulation, IEEE 802.1Q encapsulation, and multilayer switching (MLS).
Cisco IOS Mobile Wireless Gateway GPRS Support Node Configuration Guide *Cisco IOS Mobile Wireless Gateway GPRS Support Node Command Reference*	Cisco IOS Gateway GPRS Support Node (GGSN) in a 2.5-generation general packet radio service (GPRS) and 3-generation universal mobile telecommunication system (UMTS) network.
Cisco IOS Mobile Wireless Home Agent Configuration Guide *Cisco IOS Mobile Wireless Home Agent Command Reference*	Cisco Mobile Wireless Home Agent, an anchor point for mobile terminals for which mobile IP or proxy mobile IP services are provided.
Cisco IOS Mobile Wireless Packet Data Serving Node Configuration Guide *Cisco IOS Mobile Wireless Packet Data Serving Node Command Reference*	Cisco Packet Data Serving Node (PDSN), a wireless gateway that is between the mobile infrastructure and standard IP networks and that enables packet data services in a code division multiple access (CDMA) environment.
Cisco IOS Mobile Wireless Radio Access Networking Configuration Guide *Cisco IOS Mobile Wireless Radio Access Networking Command Reference*	Cisco IOS radio access network products.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Multiprotocol Label Switching Configuration Guide *Cisco IOS XE Multiprotocol Label Switching Configuration Guide* *Cisco IOS Multiprotocol Label Switching Command Reference*	MPLS Label Distribution Protocol (LDP), MPLS Layer 2 VPNs, MPLS Layer 3 VPNs, MPLS Traffic Engineering (TE), and MPLS Embedded Management (EM) and MIBs.
Cisco IOS Multi-Topology Routing Configuration Guide *Cisco IOS Multi-Topology Routing Command Reference*	Unicast and multicast topology configurations, traffic classification, routing protocol support, and network management support.
Cisco IOS NetFlow Configuration Guide *Cisco IOS XE NetFlow Configuration Guide* *Cisco IOS NetFlow Command Reference*	Network traffic data analysis, aggregation caches, export features.
Cisco IOS Network Management Configuration Guide *Cisco IOS XE Network Management Configuration Guide* *Cisco IOS Network Management Command Reference*	Basic system management; system monitoring and logging; troubleshooting, logging, and fault management; Cisco Discovery Protocol; Cisco IOS Scripting with Tool Control Language (Tcl); Cisco networking services (CNS); DistributedDirector; Embedded Event Manager (EEM); Embedded Resource Manager (ERM); Embedded Syslog Manager (ESM); HTTP; Remote Monitoring (RMON); SNMP; and VPN Device Manager Client for Cisco IOS Software (XSM Configuration).
Cisco IOS Novell IPX Configuration Guide *Cisco IOS XE Novell IPX Configuration Guide* *Cisco IOS Novell IPX Command Reference*	Novell Internetwork Packet Exchange (IPX) protocol.
Cisco IOS Optimized Edge Routing Configuration Guide *Cisco IOS Optimized Edge Routing Command Reference*	Optimized edge routing (OER) monitoring, policy configuration, routing control, logging and reporting, and VPN IPsec/generic routing encapsulation (GRE) tunnel interface optimization.
Cisco IOS Quality of Service Solutions Configuration Guide *Cisco IOS XE Quality of Service Solutions Configuration Guide* *Cisco IOS Quality of Service Solutions Command Reference*	Class-based weighted fair queuing (CBWFQ), custom queuing, distributed traffic shaping (DTS), generic traffic shaping (GTS), IP- to-ATM class of service (CoS), low latency queuing (LLQ), modular QoS CLI (MQC), Network-Based Application Recognition (NBAR), priority queuing, Security Device Manager (SDM), Multilink PPP (MLPPP) for QoS, header compression, AutoQoS, QoS features for voice, Resource Reservation Protocol (RSVP), weighted fair queuing (WFQ), and weighted random early detection (WRED).
Cisco IOS Security Configuration Guide *Cisco IOS XE Security Configuration Guide* *Cisco IOS Security Command Reference*	Access control lists (ACLs), authentication, authorization, and accounting (AAA), firewalls, IP security and encryption, neighbor router authentication, network access security, network data encryption with router authentication, public key infrastructure (PKI), RADIUS, TACACS+, terminal access security, and traffic filters.

Table 1 Cisco IOS and Cisco IOS XE Configuration Guides and Command References (continued)

Configuration Guide and Command Reference Titles	Features/Protocols/Technologies
Cisco IOS Service Selection Gateway Configuration Guide *Cisco IOS Service Selection Gateway Command Reference*	Subscriber authentication, service access, and accounting.
Cisco IOS Software Activation Configuration Guide *Cisco IOS Software Activation Command Reference*	An orchestrated collection of processes and components to activate Cisco IOS software feature sets by obtaining and validating Cisco software licenses.
Cisco IOS Software Modularity Installation and Configuration Guide *Cisco IOS Software Modularity Command Reference*	Installation and basic configuration of software modularity images, including installations on single and dual route processors, installation rollbacks, software modularity binding, software modularity processes and patches.
Cisco IOS Terminal Services Configuration Guide *Cisco IOS Terminal Services Command Reference* *Cisco IOS XE Terminal Services Command Reference*	DEC, local-area transport (LAT), and X.25 packet assembler/disassembler (PAD).
Cisco IOS Virtual Switch Command Reference	Virtual switch redundancy, high availability, and packet handling; converting between standalone and virtual switch modes; virtual switch link (VSL); Virtual Switch Link Protocol (VSLP). **Note** For information about virtual switch configuration, refer to the product-specific software configuration information for the Cisco Catalyst 6500 series switch or for the Metro Ethernet 6500 series switch.
Cisco IOS Voice Configuration Library *Cisco IOS Voice Command Reference*	Cisco IOS support for voice call control protocols, interoperability, physical and virtual interface management, and troubleshooting. The library includes documentation for IP telephony applications.
Cisco IOS VPDN Configuration Guide *Cisco IOS XE VPDN Configuration Guide* *Cisco IOS VPDN Command Reference*	Layer 2 Tunneling Protocol (L2TP) dial-out load balancing and redundancy, L2TP extended failover, L2TP security VPDN, multihop by Dialed Number Identification Service (DNIS), timer and retry enhancements for L2TP and Layer 2 Forwarding (L2F), RADIUS Attribute 82: tunnel assignment ID, shell-based authentication of VPDN users, tunnel authentication via RADIUS on tunnel terminator.
Cisco IOS Wide-Area Networking Configuration Guide *Cisco IOS XE Wide-Area Networking Configuration Guide* *Cisco IOS Wide-Area Networking Command Reference*	Frame Relay, Layer 2 Tunneling Protocol Version 3 (L2TPv3), Link Access Procedure, Balanced (LAPB), Switched Multimegabit Data Service (SMDS), and X.25.
Cisco IOS Wireless LAN Configuration Guide *Cisco IOS Wireless LAN Command Reference*	Broadcast key rotation, IEEE 802.11x support, IEEE 802.1x authenticator, IEEE 802.1x local authentication service for Extensible Authentication Protocol-Flexible Authentication via Secure Tunneling (EAP-FAST), Multiple Basic Service Set ID (BSSID), Wi-Fi Multimedia (WMM) required elements, and Wi-Fi Protected Access (WPA).

Table 2 Cisco IOS Supplementary Documents and Resources

Document Title	Description
Cisco IOS Master Command List, All Releases	Alphabetical list of all the commands documented in all Cisco IOS releases.
Cisco IOS New, Modified, Removed, and Replaced Commands	List of all the new, modified, removed, and replaced commands for a Cisco IOS release.
Cisco IOS Software System Messages	List of Cisco IOS system messages and descriptions. System messages may indicate problems with your system; be informational only; or may help diagnose problems with communications lines, internal hardware, or the system software.
Cisco IOS Debug Command Reference	Alphabetical list of **debug** commands including brief descriptions of use, command syntax, and usage guidelines.
Release Notes and Caveats	Information about new and changed features, system requirements, and other useful information about specific software releases; information about defects in specific Cisco IOS software releases.
MIBs	Files used for network monitoring. To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator at the following URL: http://www.cisco.com/go/mibs
RFCs	Standards documents maintained by the Internet Engineering Task Force (IETF) that Cisco IOS documentation references where applicable. The full text of referenced RFCs may be obtained at the following URL: http://www.rfc-editor.org/

Additional Resources and Documentation Feedback

What's New in Cisco Product Documentation is published monthly and describes all new and revised Cisco technical documentation. The *What's New in Cisco Product Documentation* publication also provides information about obtaining the following resources:

- Technical documentation
- Cisco product security overview
- Product alerts and field notices
- Technical assistance

Cisco IOS technical documentation includes embedded feedback forms where you can rate documents and provide suggestions for improvement. Your feedback helps us improve our documentation.

Using the Command-Line Interface in Cisco IOS and Cisco IOS XE Software

Last updated: August 6, 2008

This document provides basic information about the command-line interface (CLI) in Cisco IOS and Cisco IOS XE software and how you can use some of the CLI features. This document contains the following sections:

For more information about using the CLI, see the "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*.

For information about the software documentation set, see the "About Cisco IOS and Cisco IOS XE Software Documentation" document.

Initially Configuring a Device

Initially configuring a device varies by platform. For information about performing an initial configuration, see the hardware installation documentation that is provided with the original packaging of the product or go to the Product Support area of Cisco.com at http://www.cisco.com/web/psa/products/index.html.

After you have performed the initial configuration and connected the device to your network, you can configure the device by using the console port or a remote access method, such as Telnet or Secure Shell (SSH), to access the CLI or by using the configuration method provided on the device, such as Security Device Manager.

Changing the Default Settings for a Console or AUX Port

There are only two changes that you can make to a console port and an AUX port:

- Change the port speed with the **config-register 0x** command. Changing the port speed is not recommended. The well-known default speed is 9600.
- Change the behavior of the port; for example, by adding a password or changing the timeout value.

Note The AUX port on the Route Processor (RP) installed in a Cisco ASR1000 series router does not serve any useful customer purpose and should be accessed only under the advisement of a customer support representative.

Using the CLI

This section describes the following topics:

Understanding Command Modes

The CLI command mode structure is hierarchical, and each mode supports a set of specific commands. This section describes the most common of the many modes that exist.

Table 1 lists common command modes with associated CLI prompts, access and exit methods, and a brief description of how each mode is used.

Table 1 CLI Command Modes

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
User EXEC	Log in.	`Router>`	Issue the **logout** or **exit** command.	• Change terminal settings. • Perform basic tests. • Display device status.
Privileged EXEC	From user EXEC mode, issue the **enable** command.	`Router#`	Issue the **disable** command or the **exit** command to return to user EXEC mode.	• Issue **show** and **debug** commands. • Copy images to the device. • Reload the device. • Manage device configuration files. • Manage device file systems.
Global configuration	From privileged EXEC mode, issue the **configure terminal** command.	`Router(config)#`	Issue the **exit** command or the **end** command to return to privileged EXEC mode.	Configure the device.
Interface configuration	From global configuration mode, issue the **interface** command.	`Router(config-if)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual interfaces.
Line configuration	From global configuration mode, issue the **line vty** or **line console** command.	`Router(config-line)#`	Issue the **exit** command to return to global configuration mode or the **end** command to return to privileged EXEC mode.	Configure individual terminal lines.

Table 1 CLI Command Modes (continued)

Command Mode	Access Method	Prompt	Exit Method	Mode Usage
ROM monitor	From privileged EXEC mode, issue the **reload** command. Press the **Break** key during the first 60 seconds while the system is booting.	`rommon # >` The # symbol represents the line number and increments at each prompt.	Issue the **continue** command.	• Run as the default operating mode when a valid image cannot be loaded. • Access the fall-back procedure for loading an image when the device lacks a valid image and cannot be booted. • Perform password recovery when a CTRL-Break sequence is issued within 60 seconds of a power-on or reload event.
Diagnostic (available only on the Cisco ASR1000 series router)	The router boots or enters diagnostic mode in the following scenarios. When a Cisco IOS process or processes fail, in most scenarios the router will reload. • A user-configured access policy was configured using the **transport-map** command, which directed the user into diagnostic mode. • The router was accessed using an RP auxiliary port. • A break signal (**Ctrl-C**, **Ctrl-Shift-6**, or the **send break** command) was entered, and the router was configured to enter diagnostic mode when the break signal was received.	`Router(diag)#`	If a Cisco IOS process failure is the reason for entering diagnostic mode, the failure must be resolved and the router must be rebooted to exit diagnostic mode. If the router is in diagnostic mode because of a transport-map configuration, access the router through another port or using a method that is configured to connect to the Cisco IOS CLI. If the RP auxiliary port was used to access the router, use another port for access. Accessing the router through the auxiliary port is not useful for customer purposes.	• Inspect various states on the router, including the Cisco IOS state. • Replace or roll back the configuration. • Provide methods of restarting the Cisco IOS software or other processes. • Reboot hardware, such as the entire router, an RP, an ESP, a SIP, a SPA, or possibly other hardware components. • Transfer files into or off of the router using remote access methods such as FTP, TFTP, and SCP.

EXEC commands are not saved when the software reboots. Commands that you issue in a configuration mode can be saved to the startup configuration. If you save the running configuration to the startup configuration, these commands will execute when the software is rebooted. Global configuration mode is the highest level of configuration mode. From global configuration mode, you can enter a variety of other configuration modes, including protocol-specific modes.

ROM monitor mode is a separate mode that is used when the software cannot load properly. If a valid software image is not found when the software boots or if the configuration file is corrupted at startup, the software might enter ROM monitor mode. Use the question symbol (?) to view the commands that you can use while the device is in ROM monitor mode.

```
rommon 1 > ?
alias               set and display aliases command
boot                boot up an external process
confreg             configuration register utility
cont                continue executing a downloaded image
context             display the context of a loaded image
cookie              display contents of cookie PROM in hex
.
.
.
rommon 2 >
```

The following example shows how the command prompt changes to indicate a different command mode:

```
Router> enable
Router# configure terminal
Router(config)# interface ethernet 1/1
Router(config-if)# ethernet
Router(config-line)# exit
Router(config)# end
Router#
```

Note A keyboard alternative to the **end** command is Ctrl-Z.

Using the Interactive Help Feature

The CLI includes an interactive Help feature. Table 2 describes how to use the Help feature.

Table 2 CLI Interactive Help Commands

Command	Purpose
help	Provides a brief description of the help feature in any command mode.
?	Lists all commands available for a particular command mode.
*partial command***?**	Provides a list of commands that begin with the character string (no space between the command and the question mark).
partial command<**Tab**>	Completes a partial command name (no space between the command and <Tab>).
command **?**	Lists the keywords, arguments, or both associated with the command (space between the command and the question mark).
command keyword **?**	Lists the arguments that are associated with the keyword (space between the keyword and the question mark).

The following examples show how to use the help commands:

help

```
Router> help

Help may be requested at any point in a command by entering a question mark '?'.  If
nothing matches, the help list will be empty and you must backup until entering a '?'
shows the available options.

Two styles of help are provided:

1. Full help is available when you are ready to enter a command argument (e.g. 'show ?')
and describes each possible argument.

2. Partial help is provided when an abbreviated argument is entered and you want to know
what arguments match the input (e.g. 'show pr?'.)
```

?

```
Router# ?
Exec commands:
  access-enable        Create a temporary access-List entry
  access-profile       Apply user-profile to interface
  access-template      Create a temporary access-List entry
  alps                 ALPS exec commands
  archive              manage archive files
<snip>
```

***partial command*?**

```
Router(config)# zo?
zone  zone-pair
```

***partial command*<Tab>**

```
Router(config)# we<Tab> webvpn
```

***command* ?**

```
Router(config-if)# pppoe ?
  enable        Enable pppoe
  max-sessions  Maximum PPPOE sessions
```

***command keyword* ?**

```
Router(config-if)# pppoe enable ?
  group  attach a BBA group
  <cr>
```

Understanding Command Syntax

Command syntax is the format in which a command should be entered in the CLI. Commands include the name of the command, keywords, and arguments. Keywords are alphanumeric strings that are used literally. Arguments are placeholders for values that a user must supply. Keywords and arguments may be required or optional.

Specific conventions convey information about syntax and command elements. Table 3 describes these conventions.

Table 3 CLI Syntax Conventions

Symbol/Text	Function	Notes
< > (angle brackets)	Indicate that the option is an argument.	Sometimes arguments are displayed without angle brackets.
A.B.C.D.	Indicates that you must enter a dotted decimal IP address.	Angle brackets (< >) are not always used to indicate that an IP address is an argument.
WORD (all capital letters)	Indicates that you must enter one word.	Angle brackets (< >) are not always used to indicate that a WORD is an argument.
LINE (all capital letters)	Indicates that you must enter more than one word.	Angle brackets (< >) are not always used to indicate that a LINE is an argument.
<cr> (carriage return)	Indicates the end of the list of available keywords and arguments, and also indicates when keywords and arguments are optional. When <cr> is the only option, you have reached the end of the branch or the end of the command if the command has only one branch.	—

The following examples show syntax conventions:

```
Router(config)# ethernet cfm domain ?
  WORD  domain name
Router(config)# ethernet cfm domain dname ?
  level
Router(config)# ethernet cfm domain dname level ?
  <0-7>  maintenance level number
Router(config)# ethernet cfm domain dname level 7 ?
  <cr>
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
Router(config)# logging host ?
  Hostname or A.B.C.D  IP address of the syslog server
  ipv6                 Configure IPv6 syslog server
Router(config)# snmp-server file-transfer access-group 10 ?
  protocol  protocol options
  <cr>
```

Understanding Enable and Enable Secret Passwords

Some privileged EXEC commands are used for actions that impact the system, and it is recommended that you set a password for these commands to prevent unauthorized use. Two types of passwords, enable (not encrypted) and enable secret (encrypted), can be set. The following commands set these passwords and are issued in global configuration mode:

- **enable** *password*
- **enable secret** *password*

Using an enable secret password is recommended because it is encrypted and more secure than the enable password. When you use an enable secret password, text is encrypted (unreadable) before it is written to the config.text file. When you use an enable password, the text is written as entered (readable) to the config.text file.

Each type of password is case sensitive, can contain from 1 to 25 uppercase and lowercase alphanumeric characters, and can start with a number. Spaces are also valid password characters; for example, "two words" is a valid password. Leading spaces are ignored, but trailing spaces are recognized.

Note Both password commands have numeric keywords that are single integer values. If you choose a number for the first character of your password followed by a space, the system will read the number as if it were the numeric keyword and not as part of your password.

When both passwords are set, the enable secret password takes precedence over the enable password.

To remove a password, use the **no** form of the commands: **no enable** *password* or **no enable secret** *password*.

For more information about password recovery procedures for Cisco products, see http://www.cisco.com/en/US/products/sw/iosswrel/ps1831/products_tech_note09186a00801746e6.shtml.

Using the Command History Feature

The CLI command history feature saves the commands you enter during a session in a command history buffer. The default number of commands saved is 10, but the number is configurable within the range of 0 to 256. This command history feature is particularly useful for recalling long or complex commands.

To change the number of commands saved in the history buffer for a terminal session, issue the **terminal history size** command:

```
Router# terminal history size num
```

A command history buffer is also available in line configuration mode with the same default and configuration options. To set the command history buffer size for a terminal session in line configuration mode, issue the **history** command:

```
Router(config-line)# history [size num]
```

To recall commands from the history buffer, use the following methods:

- Press Ctrl-P or the up arrow key—Recalls commands beginning with the most recent command. Repeat the key sequence to recall successively older commands.

- Press Ctrl-N or the down arrow key—Recalls the most recent commands in the history buffer after they have been recalled using Ctrl-P or the up arrow key. Repeat the key sequence to recall successively more recent commands.

Note The arrow keys function only on ANSI-compatible terminals such as the VT100.

- Issue the **show history** command in user EXEC or privileged EXEC mode—Lists the most recent commands that you entered. The number of commands that are displayed is determined by the setting of the **terminal history size** and **history** commands.

  The CLI command history feature is enabled by default. To disable this feature for a terminal session, issue the **terminal no history** command in user EXEC or privileged EXEC mode or the **no history** command in line configuration mode.

Abbreviating Commands

Typing a complete command name is not always required for the command to execute. The CLI recognizes an abbreviated command when the abbreviation contains enough characters to uniquely identify the command. For example, the **show version** command can be abbreviated as **sh ver**. It cannot be abbreviated as **s ver** because **s** could mean **show**, **set**, or **systat**. The **sh v** abbreviation also is not valid because the **show** command has **vrrp** as a keyword in addition to **version**. (Command and keyword examples from Cisco IOS Release 12.4(13)T.)

Using Aliases for CLI Commands

To save time and the repetition of entering the same command multiple times, you can use a command alias. An alias can be configured to do anything that can be done at the command line, but an alias cannot move between modes, type in passwords, or perform any interactive functions.

Table 4 shows the default command aliases.

Table 4 Default Command Aliases

Command Alias	Original Command
h	help
lo	logout
p	ping
s	show
u or **un**	undebug
w	where

To create a command alias, issue the **alias** command in global configuration mode. The syntax of the command is **alias** *mode command-alias original-command*. Following are some examples:

- Router(config)# **alias exec prt partition**—privileged EXEC mode
- Router(config)# **alias configure sb source-bridge**—global configuration mode
- Router(config)# **alias interface rl rate-limit**—interface configuration mode

To view both default and user-created aliases, issue the **show alias** command.

For more information about the **alias** command, see http://www.cisco.com/en/US/docs/ios/fundamentals/command/reference/cf_book.html.

Using the no and default Forms of Commands

Most configuration commands have a **no** form that is used to reset a command to its default value or disable a feature or function. For example, the **ip routing** command is enabled by default. To disable this command, you would issue the **no ip routing** command. To re-enable IP routing, you would issue the **ip routing** command.

Configuration commands may also have a **default** form, which returns the command settings to their default values. For commands that are disabled by default, using the **default** form has the same effect as using the **no** form of the command. For commands that are enabled by default and have default settings, the **default** form enables the command and returns the settings to their default values.

The **no** and **default** forms of commands are described in the command pages of command references.

Using the debug Command

A **debug** command produces extensive output that helps you troubleshoot problems in your network. These commands are available for many features and functions within Cisco IOS and Cisco IOS XE software. Some **debug** commands are **debug all**, **debug aaa accounting**, and **debug mpls packets**. To use **debug** commands during a Telnet session with a device, you must first enter the **terminal monitor** command. To turn off debugging completely, you must enter the **undebug all** command.

For more information about **debug** commands, see the *Cisco IOS Debug Command Reference* at http://www.cisco.com/en/US/docs/ios/debug/command/reference/db_book.html.

Caution Debugging is a high priority and high CPU utilization process that can render your device unusable. Use **debug** commands only to troubleshoot specific problems. The best times to run debugging are during periods of low network traffic and when few users are interacting with the network. Debugging during these periods decreases the likelihood that the **debug** command processing overhead will affect network performance or user access or response times.

Filtering Output Using Output Modifiers

Many commands produce lengthy output that may use several screens to display. Using output modifiers, you can filter this output to show only the information that you want to see.

Three output modifiers are available and are described as follows:

- **begin** *regular expression*—Displays the first line in which a match of the regular expression is found and all lines that follow.
- **include** *regular expression*—Displays all lines in which a match of the regular expression is found.
- **exclude** *regular expression*—Displays all lines except those in which a match of the regular expression is found.

To use one of these output modifiers, type the command followed by the pipe symbol (|), the modifier, and the regular expression that you want to search for or filter. A regular expression is a case-sensitive alphanumeric pattern. It can be a single character or number, a phrase, or a more complex string.

The following example illustrates how to filter output of the **show interface** command to display only lines that include the expression "protocol."

```
Router# show interface | include protocol

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

Understanding CLI Error Messages

You may encounter some error messages while using the CLI. Table 5 shows the common CLI error messages.

Table 5 Common CLI Error Messages

Error Message	Meaning	How to Get Help
% Ambiguous command: "show con"	You did not enter enough characters for the command to be recognized.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Incomplete command.	You did not enter all the keywords or values required by the command.	Reenter the command followed by a space and a question mark (?). The keywords that you are allowed to enter for the command appear.
% Invalid input detected at "^" marker.	You entered the command incorrectly. The caret (^) marks the point of the error.	Enter a question mark (?) to display all the commands that are available in this command mode. The keywords that you are allowed to enter for the command appear.

For more system error messages, see the following documents:

- *Cisco IOS Release 12.2SR System Message Guide*
- *Cisco IOS System Messages, Volume 1 of 2* (Cisco IOS Release 12.4)
- *Cisco IOS System Messages, Volume 2 of 2* (Cisco IOS Release 12.4)

Saving Changes to a Configuration

To save changes that you made to the configuration of a device, you must issue the **copy running-config startup-config** command or the **copy system:running-config nvram:startup-config** command. When you issue these commands, the configuration changes that you made are saved to the startup configuration and saved when the software reloads or power to the device is turned off or interrupted. The following example shows the syntax of the **copy running-config startup-config** command:

```
Router# copy running-config startup-config
Destination filename [startup-config]?
```

You press Enter to accept the startup-config filename (the default), or type a new filename and then press Enter to accept that name. The following output is displayed indicating that the configuration was saved:

```
Building configuration...
[OK]
Router#
```

On most platforms, the configuration is saved to NVRAM. On platforms with a Class A flash file system, the configuration is saved to the location specified by the CONFIG_FILE environment variable. The CONFIG_FILE variable defaults to NVRAM.

Additional Information

- "Using the Cisco IOS Command-Line Interface" section of the *Cisco IOS Configuration Fundamentals Configuration Guide*:

  http://www.cisco.com/en/US/docs/ios/fundamentals/configuration/guide/cf_cli-basics.html

  or

  "Using Cisco IOS XE Software" chapter of the *Cisco ASR1000 Series Aggregation Services Routers Software Configuration Guide*:

  http://www.cisco.com/en/US/docs/routers/asr1000/configuration/guide/chassis/using_cli.html
- Cisco Product Support Resources

  http://www.cisco.com/web/psa/products/index.html
- Support area on Cisco.com (also search for documentation by task or product)

  http://www.cisco.com/en/US/support/index.html
- *White Paper: Cisco IOS Reference Guide*

  http://www.cisco.com/en/US/products/sw/iosswrel/ps1828/products_white_paper09186a008018305e.shtml
- Software Download Center (downloads; tools; licensing, registration, advisory, and general information) (requires Cisco.com User ID and password)

  http://www.cisco.com/kobayashi/sw-center/
- Error Message Decoder, a tool to help you research and resolve error messages for Cisco IOS software

  http://www.cisco.com/pcgi-bin/Support/Errordecoder/index.cgi

- Command Lookup Tool, a tool to help you find detailed descriptions of Cisco IOS commands (requires Cisco.com user ID and password)

  http://tools.cisco.com/Support/CLILookup

- Output Interpreter, a troubleshooting tool that analyzes command output of supported **show** commands

  https://www.cisco.com/pcgi-bin/Support/OutputInterpreter/home.pl\

Dial Interfaces, Controllers, and Lines

Configuring Asynchronous Lines and Interfaces

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.

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 Configuration and Management

Modem Signal and Line States

This chapter describes modem states in the following section:

- Signal and Line State Diagrams

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

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

Signal and Line State Diagrams

The following signal and line state diagrams accompany some of the tasks in the following sections to illustrate how the modem control works:

- Configuring Automatic Dialing
- Automatically Answering a Modem
- Supporting Dial-In and Dial-Out Connections
- Configuring a Line Timeout Interval
- Closing Modem Connections
- Configuring a Line to Disconnect Automatically
- Supporting Reverse Modem Connections and Preventing Incoming Calls

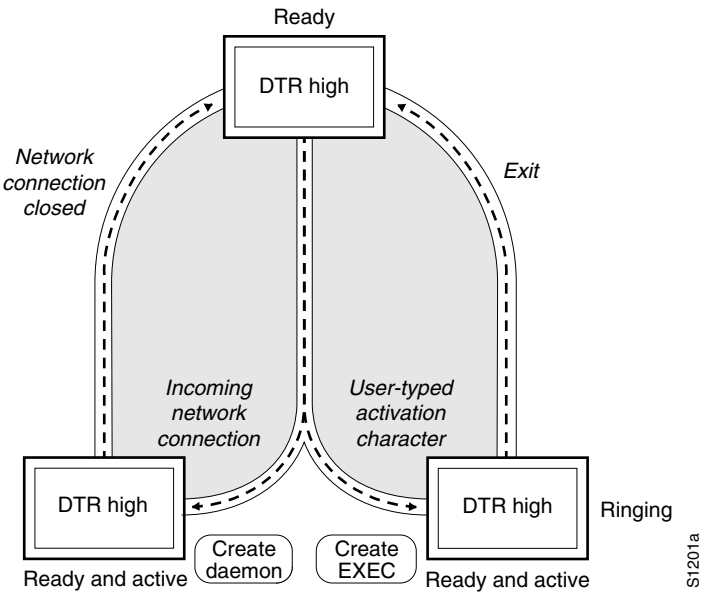
The diagrams show two processes:

- The “create daemon” process creates a tty daemon that handles the incoming network connection.
- The “create EXEC” process creates the process that interprets user commands. (See Figure 27 through Figure 31.)

In the diagrams, the current signal state and the signal the line is watching are listed inside each box. The state of the line (as displayed by the **show line** EXEC command) is listed next to the box. Events that change that state appear in italics along the event path, and actions that the software performs are described within ovals.

Figure 27 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 27 EXEC and Daemon Creation on a Line with No Modem Control



Configuring Automatic Dialing

With the dialup capability, you can set a modem to dial the phone number of a remote router automatically. This feature offers cost savings because phone line connections are made only when they are needed—you pay for using the phone line only when there is data to be received or sent.

To configure a line for automatic dialing, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem dtr-active`	Configures a line to initiate automatic dialing.

Using the **modem dtr-active** command causes a line to raise DTR signal only when there is an outgoing connection (such as reverse Telnet, NetWare Asynchronous Support Interface (NASI), or DDR), rather than leave DTR raised all the time. When raised, DTR potentially tells the modem that the router is ready to accept a call.

Automatically Answering a Modem

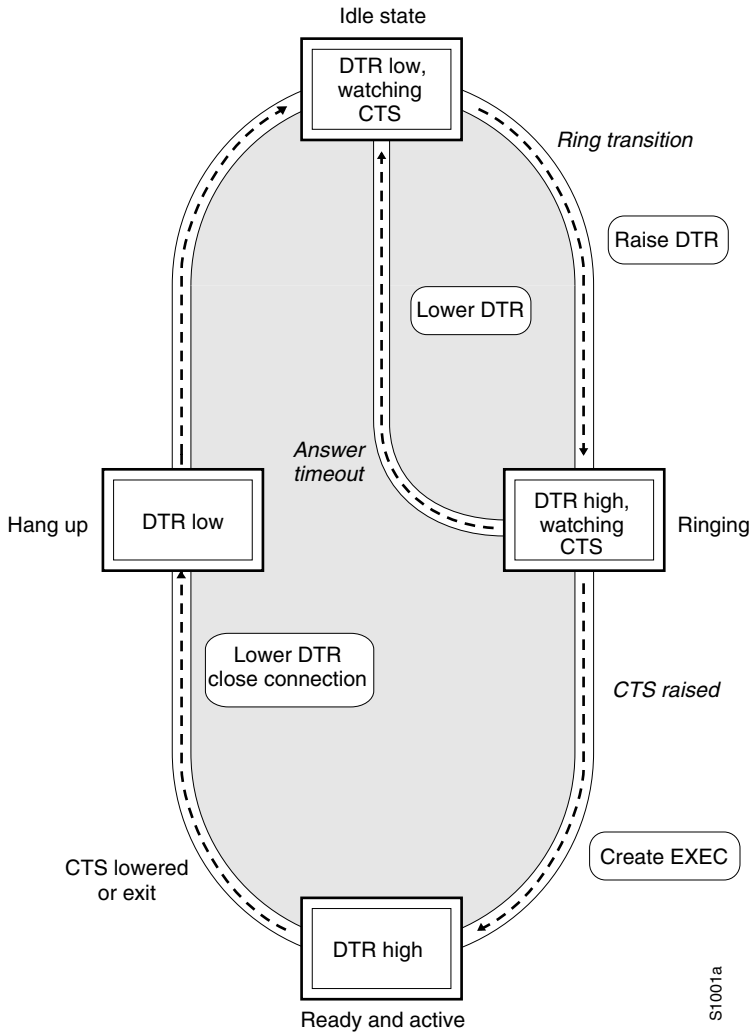
You can configure a line to answer a modem automatically. You also can configure the modem to answer the telephone on its own (as long as DTR is high), drop connections when DTR is low, and use its Carrier Detect (CD) signal to accurately reflect the presence of carrier. (Configuring the modem is a modem-dependent process.) First, wire the modem CD signal (generally pin-8) to the router RING input (pin-22), then use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem dialin`	Configures a line to automatically answer a modem.

You can turn on modem hardware flow control independently to respond to the status of router CTS input. Wire CTS to whatever signal the modem uses for hardware flow control. If the modem expects to control hardware flow in both directions, you might also need to wire modem flow control input to some other signal that the router always has high, such as the DTR signal.

Figure 28 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 28 EXEC Creation on a Line Configured for a High-Speed Modem



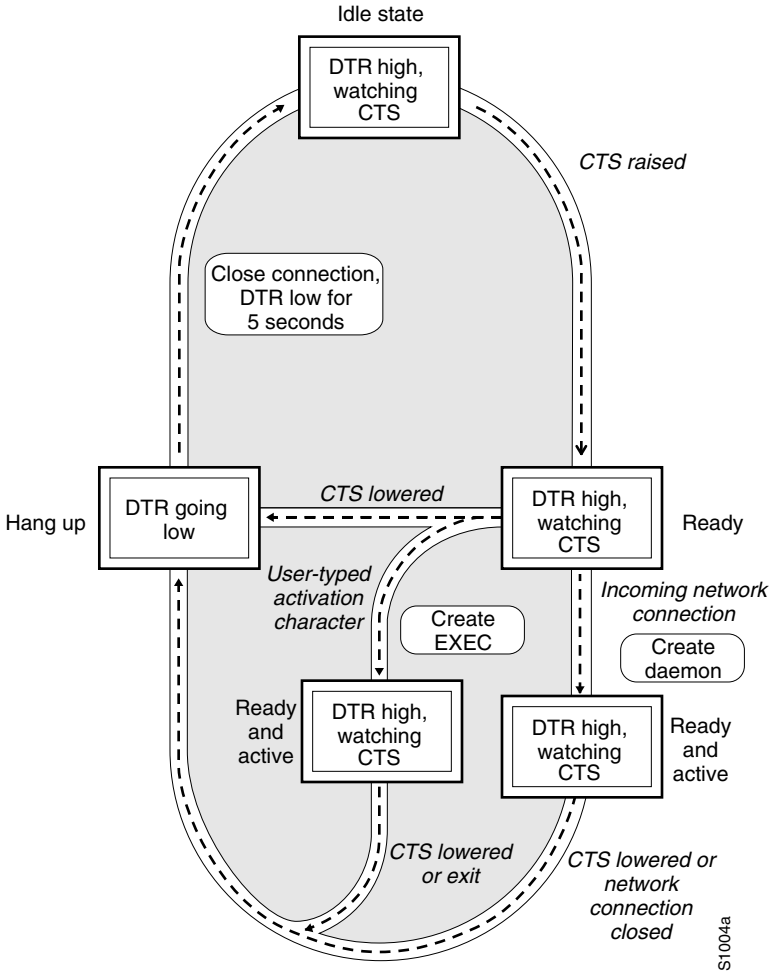
Supporting Dial-In and Dial-Out Connections

To configure a line for both incoming and outgoing calls, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem inout`	Configures a line for both incoming and outgoing calls.

Figure 29 illustrates the **modem inout** command. If the line is activated by raising the data set ready (DSR) signal, it functions exactly as a line configured with the **modem dialin** line configuration command described in the section "Automatically Answering a Modem" earlier in this chapter. If the line is activated by an incoming TCP connection, the line functions similarly to lines not used with modems.

Figure 29 EXEC and Daemon Creation for Incoming and Outgoing Calls



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

Configuring a Line Timeout Interval

To change the interval that the Cisco IOS software waits for the CTS signal after raising the DTR signal in response to the DSR (the default is 15 seconds), use the following command in line configuration mode. The timeout applies to the **modem callin** command only.

Command	Purpose
`Router(config-line)# modem answer-timeout` *seconds*	Configures modem line timing.

Note The DSR signal is called RING on older ASM-style chassis.

Closing Modem Connections

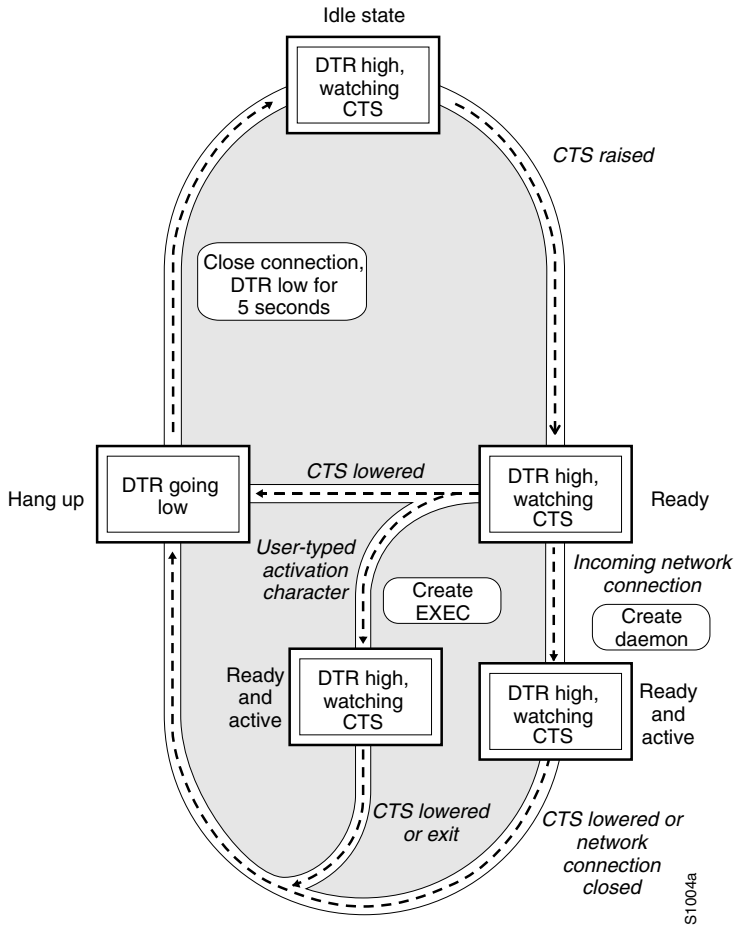
Note The **modem cts-required** command was replaced by the **modem printer** command in Cisco IOS Release 12.2.

To configure a line to close connections from a user's terminal when the terminal is turned off and to prevent inbound connections to devices that are out of service, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# modem cts-required`	Configures a line to close connections.

Figure 30 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 30 EXEC and Daemon Creation on a Line Configured for Continuous CTS



Configuring a Line to Disconnect Automatically

To configure automatic line disconnect, use the following command in line configuration mode:

Command	Purpose
`Router(config-line)# autohangup`	Configures automatic line disconnect.

The **autohangup** command causes the EXEC facility to issue the **exit** command when the last connection closes. This feature is useful for UNIX-to-UNIX copy program (UUCP) applications because UUCP scripts cannot issue a command to hang up the telephone. This feature is not used often.

Supporting Reverse Modem Connections and Preventing Incoming Calls

In addition to initiating connections, the Cisco IOS software can receive incoming connections. This capability allows you to attach serial and parallel printers, modems, and other shared peripherals to the router or access server and drive them remotely from other modem-connected systems. The Cisco IOS software supports reverse TCP, XRemote, and local-area transport (LAT) connections.

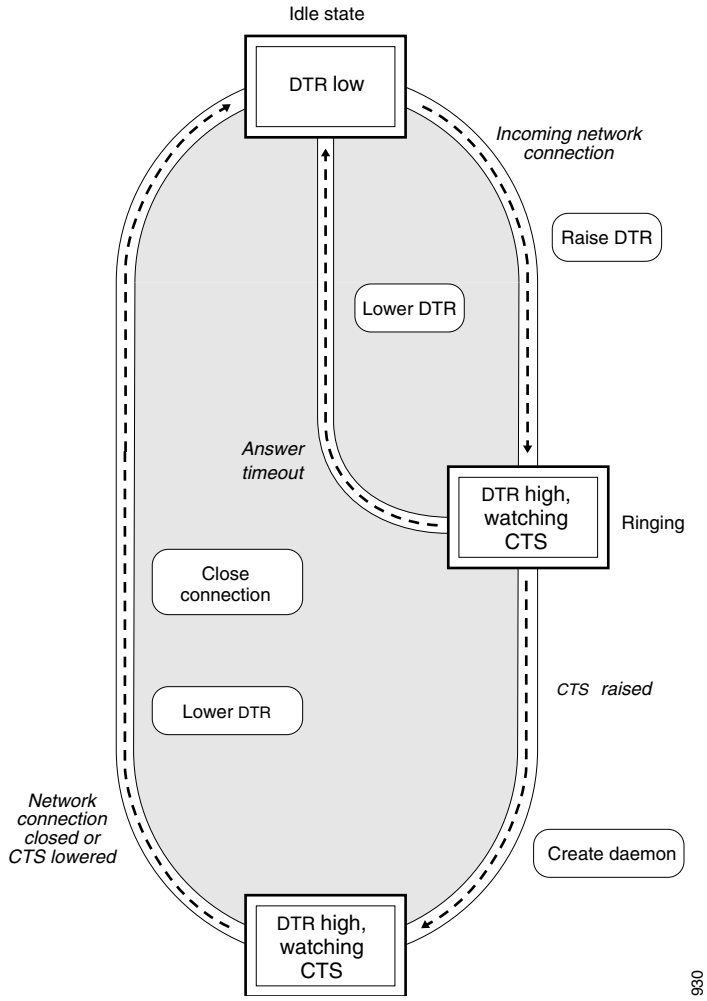
The specific TCP port or socket to which you attach the device determines the type of service that the Cisco IOS software provides on a line. When you attach the serial lines of a computer system or a data terminal switch to the serial lines of the access server, the access server can act as a network front-end device for a host that does not support the TCP/IP protocols. This arrangement is sometimes called *front-ending* or *reverse connection mode*.

The Cisco IOS software supports ports connected to computers that are connected to modems. To configure the Cisco IOS software to function somewhat like a modem, use the following command in line configuration mode. This command also prevents incoming calls.

Command	Purpose
`Router(config-line)# modem callout`	Configures a line for reverse connections and prevents incoming calls.

Figure 31 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 31 Daemon Creation on a Line Configured for Modem Dial-Out



Dial-on-Demand Routing Configuration

Configuring Snapshot Routing

This chapter describes how to configure snapshot routing. It includes the following main sections:

- Snapshot Routing Overview
- How to Configure Snapshot Routing
- Monitoring and Maintaining DDR Connections and Snapshot Routing
- Configuration Examples for Snapshot Routing

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

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

Snapshot Routing Overview

Snapshot routing enables a single router interface to call other routers during periods when the line protocol for the interface is up (these are called "active periods"). The router dials in to all configured locations during such active periods to get routes from all the remote locations.

The router can be configured to exchange routing updates each time the line protocol goes from "down" to "up" or from "dialer spoofing" to "fully up." The router can also be configured to dial the server router in the absence of regular traffic if the active period time expires.

Snapshot routing is useful in two command situations:

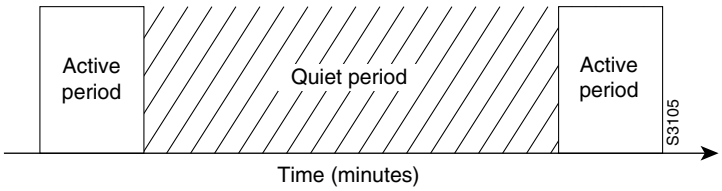
- Configuring static routes for dial-on-demand routing (DDR) interfaces
- Reducing the overhead of periodic updates sent by routing protocols to remote branch offices over a dedicated serial line

When configuring snapshot routing, you choose one router on the interface to be the client router and one or more other routers to be server routers. The client router determines the frequency at which routing information is exchanged between routers.

Routing information is exchanged during an active period. During the active period, a client router dials all the remote server routers for which it has a snapshot dialer map defined in order to get routes from all the remote locations. The server router provides information about routes to each client router that calls.

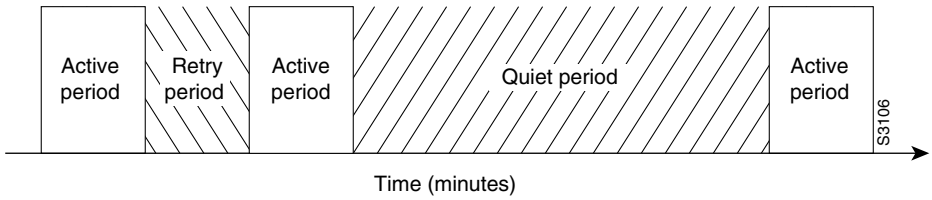
At the end of the active period, the router takes a snapshot of the entries in the routing table. These entries remain frozen during a quiet period. At the end of the quiet period, another active period starts during which routing information is again exchanged; see Figure 4.

Figure 4 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 5.

Figure 5 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
```

Dial-Backup Configuration

Reliable Static Routing Backup Using Object Tracking

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 Point-to-Point 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.

Feature History for Reliable Static Routing Backup Using Object Tracking

Release	Modification
12.3(2)XE	This feature was introduced.
12.3(8)T	Support for this feature was integrated into Cisco IOS Release 12.3(8)T.
12.3(14)T	The Cisco IOS command-line interface (CLI) used to configure the Cisco IOS IP Service Level Agreements (SLAs) monitoring and management feature set was modified.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.

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 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.

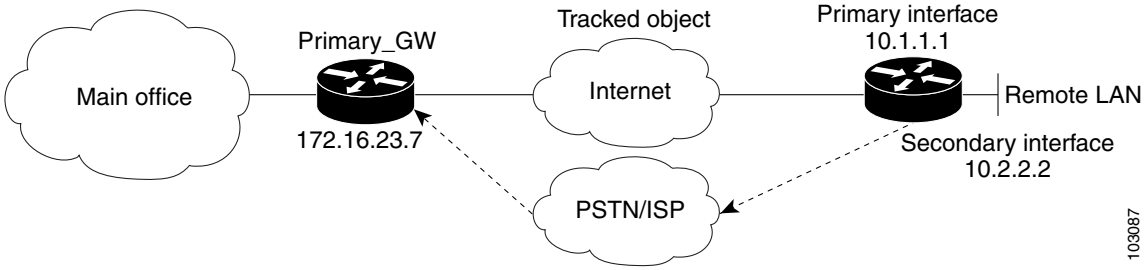
Information About Reliable Static Routing Backup Using Object Tracking

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 6.

Figure 6 Reliable Static Routing Backup Using Object Tracking Network Diagram



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 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 88 (required)
- Configuring the Backup Interface for Reliable Static Routing Backup Using Object Tracking, page 91 (required)
- Configuring Network Monitoring with Cisco IOS IP SLAs for Reliable Static Routing Backup Using Object Tracking, page 92 (required)
- Configuring the Routing Policy for Reliable Static Routing Backup Using Object Tracking, page 98 (required)
- Configuring the Default Route for the Primary Interface Using Static Routing, page 105 (required)
- Configuring a Floating Static Default Route on the Secondary Interface, page 106 (required)
- Verifying the State of the Tracked Object for Reliable Static Routing Backup Using Object Tracking, page 106 (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 unsure 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 88
- Configuring the Primary Interface for DHCP, page 89
- Configuring the Primary Interface for Static Routing, page 90

Configuring the Primary Interface for PPPoE

Perform this task to configure the primary interface if you are using 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**]

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 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 an interface configuration.
Step 5	**no ip address** **Example:** `Router(config-if)# no ip address`	Sets a primary or secondary IP address for an 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.

Configuring the Primary Interface for DHCP

Perform this task to configure the primary interface if you are using 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**

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 an 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.
Step 6	**ip address dhcp** **Example:** `Router(config-if)# ip address dhcp`	Acquires an IP address on an Ethernet interface from DHCP.

Configuring the Primary Interface for Static Routing

Perform this task to configure the primary interface if you are using 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**]

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 an interface configuration.
Step 5	**ip address** *ip-address mask* [**secondary**] **Example:** `Router(config-if)# ip address 10.1.1.1 255.0.0.0`	Sets a primary or secondary IP address for an interface.

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**]

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.
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 10.2.2.2 255.0.0.0`	Sets a primary or secondary IP address for an interface.

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 CLI 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, and 12.2(33)SRA, page 93
- 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 94
- Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.4(4)T and Later Releases, page 96

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

Perform this task to create an IP SLAs configuration to ping the target address. 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**}

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.

	Command or Action	Purpose
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.
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.

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 IP SLAs configuration 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 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**}

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.
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.

	Command or Action	Purpose
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.
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.

Configuring Cisco IOS IP SLAs for Cisco IOS Release 12.4(4)T and Later Releases

Perform this task to create an IP SLAs configuration 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**}

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** [*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.
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.

	Command or Action	Purpose
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 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.

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 98
- Configuring a Routing Policy for DHCP, page 100
- Configuring a Routing Policy for Static Routing, page 101

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*

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 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 will be routed to the null interface and discarded. Otherwise policy routing will fail and the packet will be routed using the Routing Information Base (RIB) card. Routing the packet using the RIB card is undesirable.

	Command or Action	Purpose
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.

Configuring a Routing Policy for DHCP

Perform this task to configure a routing policy if 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*

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 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.

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 101
- Configuring a Routing Policy for Static Routing with a Multipoint Primary Gateway, page 103

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*

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 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.
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.

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*

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 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.

	Command or Action	Purpose
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.

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*]

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.

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: PPPoE Example

The following example configures 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, and 12.2(33)SXH.

```
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: DHCP Example

The following example configures 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 and later releases.

```
!
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 10.2.2.2 255.0.0.0
!
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: Static Routing Examples

The following example configures 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 10.1.1.1 255.0.0.0

interface Dialer 0
```

```
 description backup-link
 ip address 10.2.2.2 255.0.0.0

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 10.1.1.1 255.0.0.0

interface ethernet 1
 description backup-link
 ip address 10.2.2.2 255.0.0.0

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	The "Configuring IPSec Network Security" chapter in the *Cisco IOS Security Configuration Guide*
IPSec commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	The *Cisco IOS Security Command Reference*
VPDN configuration tasks	The *Cisco IOS VPDN Configuration Guide*
VPDN commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	The *Cisco IOS VPDN Command Reference*
PPPoE configuration tasks	The "Configuring Broadband Access: PPP and Routed Bridge Encapsulation" chapter in the *Cisco IOS Wide-Area Networking Configuration Guide*
PPPoE commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	The *Cisco IOS Wide-Area Networking Command Reference*
DDR configuration tasks	• The "Dial-on-Demand Routing Configuration" part in the *Cisco IOS Dial Technologies Configuration Guide* • *Configuring and Troubleshooting DDR Backup*
DDR commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	The *Cisco IOS Dial Technologies Command Reference*
IP SLAs configuration tasks	*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. Access to most tools on the Cisco Support website requires a Cisco.com user ID and password. If you have a valid service contract but do not have a user ID or password, you can register on Cisco.com.	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*.

- **ip dhcp client**
- **ip route**
- **set ip next-hop dynamic dhcp**
- **show ip route track-table**

Dial-Related Addressing Services

Configuring Cisco Easy IP

This chapter describes how to configure the Cisco Easy IP feature. It includes the following main sections:

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.

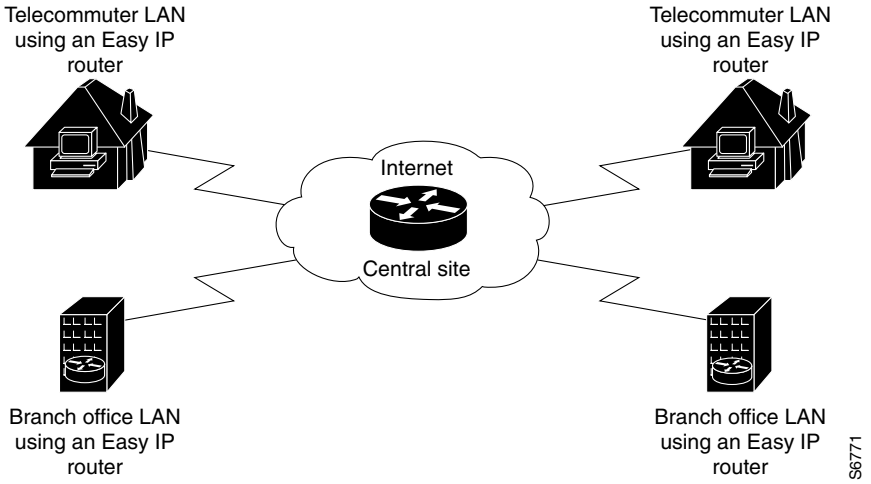
Cisco Easy IP provides the following benefits:

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

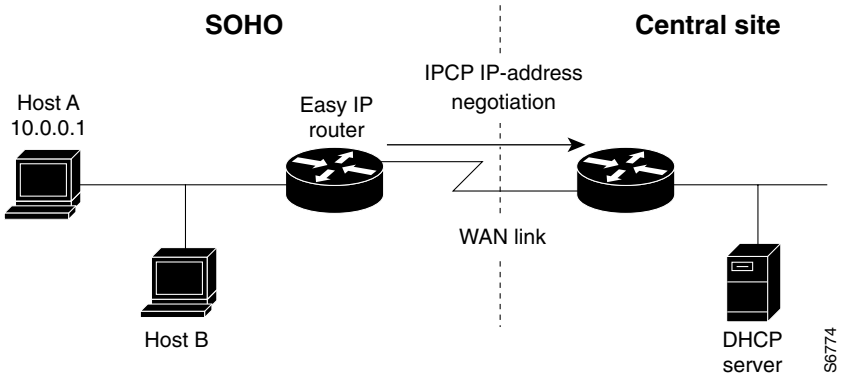
Figure 9 Telecommuter and Branch Office LANs Using Cisco Easy IP



Steps 1 through 4 show how Cisco Easy IP works:

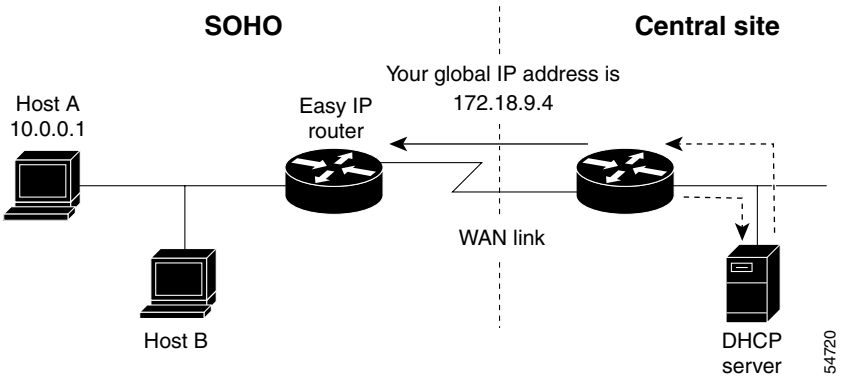
Step 1 When a SOHO host generates "interesting" traffic (as defined by Access Control Lists) for dialup (first time only), the Easy IP router requests a single registered IP address from the access server at the central site via PPP/IPCP. (See Figure 10.)

Figure 10 Cisco Easy IP Router Requests a Dynamic Global IP Address



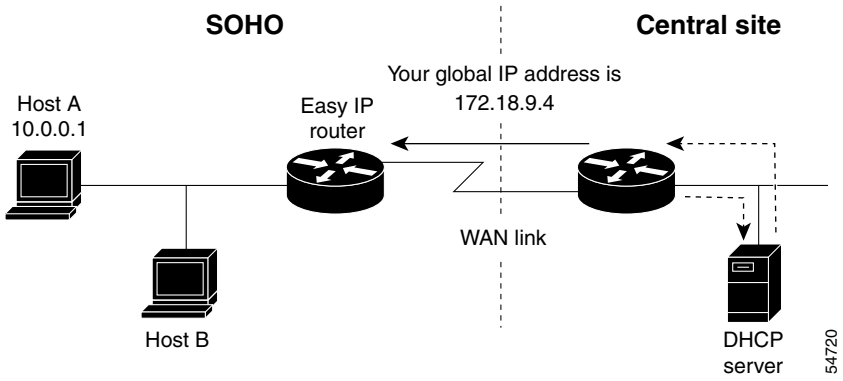
Step 2 The central site router replies with a dynamic global address from a local DHCP IP address pool. (See Figure 11.)

Figure 11 Dynamic Global IP Address Delivered to the Cisco Easy IP Router



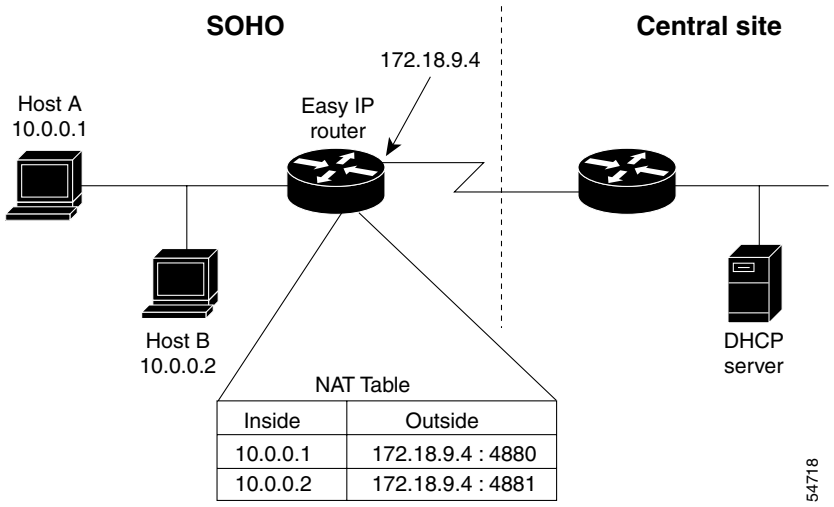
Step 3 The Cisco Easy IP router uses port-level NAT functionality to automatically create a translation that associates the registered IP address of the WAN interface with the private IP address of the client. (See Figure 12.)

Figure 12 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 13.)

Figure 13 Multiple Private Internal IP Addresses Bound to a Single Global IP Address



How to Configure Cisco Easy IP

Before using Cisco Easy IP, perform the following tasks:

- Configure the ISDN switch type and service provider identifier (SPID), if using ISDN.
- Configure the static route from LAN to WAN interface.
- Configure the Cisco IOS DHCP server.

For information about configuring ISDN switch types, see the chapter "Setting Up ISDN Basic Rate Service" earlier in this publication. For information about configuring static routes, refer to the chapter "Configuring IP Services" in the *Cisco IOS IP Configuration Guide*.

The Cisco IOS DHCP server supports both DHCP and BOOTP clients and supports finite and infinite address lease periods. DHCP address binding information is stored on a remote host via remote copy protocol (RCP), FTP, or TFTP. Refer to the *Cisco IOS IP Configuration Guide* for DHCP configuration instructions.

In its most simple configuration, a Cisco Easy IP router or access server will have a single LAN interface and a single WAN interface. Based on this model, to use Cisco Easy IP you must perform the tasks in the following sections:

- Defining the NAT Pool (Required)
- Configuring the LAN Interface (Required)
- Defining NAT for the LAN Interface (Required)
- Configuring the WAN Interface (Required)
- Enabling PPP/IPCP Negotiation (Required)
- Defining NAT for the Dialer Interface (Required)
- Configuring the Dialer Interface (Required)

For configuration examples, see the section "Configuration Examples for Cisco Easy IP" at the end of this chapter.

Defining the NAT Pool

The first step in enabling Cisco Easy IP is to create a pool of internal IP addresses to be translated. To define the NAT pool, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# access-list access-list-number permit source [source-wildcard]`	Defines a standard access list permitting those addresses that are to be translated.
Step 2	`Router(config)# ip nat inside source list access-list-number interface dialer-name overload`	Establishes dynamic source translation, identifying the access list defined in the prior step.

For information about creating access lists, refer to the chapter "Configuring IP Services" in the *Cisco IOS IP Configuration Guide*.

Configuring the LAN Interface

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

	Command	Purpose
Step 1	`Router(config)# interface type number`	Selects a specific LAN interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address address mask`	Defines the IP address and subnet mask for this interface.

For information about assigning IP addresses and subnet masks to network interfaces, refer to the chapter "Configuring IP Services" in the *Cisco IOS IP Configuration Guide*.

Defining NAT for the LAN Interface

To ensure that the LAN interface is connected to the inside network (and therefore subject to NAT), use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ip nat inside`	Defines the interface as internal for NAT.

Configuring the WAN Interface

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

	Command	Purpose
Step 1	`Router(config)# interface type number`	Selects the WAN interface and begins interface configuration mode.
Step 2	`Router(config-if)# no ip address`	Removes any associated IP address from this interface.

	Command	Purpose
Step 3	`Router(config-if)# encapsulation ppp`	Selects PPP as the encapsulation method for this interface.
Step 4	`Router(config-if)# dialer pool-member` *number*	Binds the WAN interface to the dialer interface.

Enabling PPP/IPCP Negotiation

To enable PPP/IPCP negotiation on the dialer interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *dialer-name*	Selects the dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address negotiated`	Enables PPP/IPCP negotiation for this interface.

Defining NAT for the Dialer Interface

To define that the dialer interface is connected to the outside network, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *dialer-name*	Selects the dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip nat outside`	Defines the interface as external for network address translation.

Configuring the Dialer Interface

To configure the dialer interface information, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface` *dialer-name*	Selects the dialer interface and begins interface configuration mode.
Step 2	`Router(config-if)# dialer wait-for-carrier-time` *seconds*	Specifies for a dialer interface the length of time the interface waits for a carrier before timing out.
Step 3	`Router(config-if)# dialer hold-queue` *packets*	Creates a dialer hold queue and specifies the number of packets to be held in it.
Step 4	`Router(config-if)# dialer remote-name` *username*	Specifies the remote router Challenge Handshake Authentication Protocol (CHAP) authentication name.

	Command	Purpose
Step 5	`Router(config-if)# dialer idle-timeout seconds`	Specifies the amount of idle time that can pass before calls to the central access server are disconnected. See the next section "Timeout Considerations," for more details on this setting.
Step 6	`Router(config-if)# dialer string dialer-string`	Specifies the telephone number required to reach the central access server.
Step 7	`Router(config-if)# dialer pool number`	Specifies the dialing pool to use.
Step 8	`Router(config-if)# dialer-group group-number`	Assigns the dialer interface to a dialer group.

Timeout Considerations

Dynamic NAT translations time out automatically after a predefined default period. Although configurable, with the port-level NAT functionality in Cisco Easy IP, Domain Name System (DNS) User Datagram Protocol (UDP) translations time out after 5 minutes, while DNS translations time out after 1 minute by default. TCP translations time out after 24 hours by default, unless a TCP Reset (RST) or TCP Finish (FIN) is seen in the TCP stream, in which case the translation times out after 1 minute.

If the Cisco IOS Easy IP router exceeds the dialer idle-timeout period, it is expected that all active TCP sessions were previously closed via an RST or FIN. NAT times out all TCP translations before the Cisco Easy IP router exceeds the dialer idle-timeout period. The router then renegotiates another registered IP address the next time the WAN link is brought up, thereby creating new dynamic NAT translations that bind the IP addresses of the LAN host to the newly negotiated IP address.

Configuration Examples for Cisco Easy IP

The following example shows how to configure BRI interface 0 (shown as interface bri0) to obtain its IP address via PPP/IPCP address negotiation:

```
! The following command defines the NAT pool.
ip nat inside source list 101 interface dialer1 overload
!
! The following commands define the ISDN switch type.
isdn switch type vn3
isdn tei-negotiation first-call
!
! The following commands define the LAN address and subnet mask.
interface ethernet0
 ip address 10.0.0.4 255.0.0.0

! The following command defines ethernet0 as internal for NAT.
 ip nat inside
!
! The following commands binds the physical interface to the dialer1 interface.
interface bri0
 no ip address
 encapsulation ppp
 dialer pool-member 1
!
interface dialer1
!
! The following command enables PPP/IPCP negotiation for this interface.
 ip address negotiated
 encapsulation ppp
```

```
!
! The following command defines interface dialer1 as external for NAT.
 ip nat outside
 dialer remote-name dallas
 dialer idle-timeout 180
!
! The following command defines the dialer string for the central access server.
 dialer string 4159991234
 dialer pool 1
 dialer-group 1
!
! The following commands define the static route to the WAN interface.
ip route 0.0.0.0 0.0.0.0 dialer1
access-list 101 permit ip 10.0.0.0 0.255.255.255 any
dialer-list 1 protocol ip list 101
```

The following example shows how to configure an asynchronous interface (interface async1) to obtain its IP address via PPP/IPCP address negotiation:

```
! This command defines the NAT pool.
ip nat inside source list 101 interface dialer 1 overload
!
! The following commands define the LAN IP address and subnet mask.
interface ethernet0
 ip address 10.0.0.4 255.0.0.0
!
! The following command defines ethernet0 as internal for NAT.
 ip nat inside
!
! The following commands bind the physical dialer1 interface.
interface async1
 no ip address
 encapsulation ppp
 async mode dedicated
 dialer pool-member 1
!
interface dialer1
!
! The following command enables PPP/IPCP negotiation for this interface.
 ip address negotiated
 encapsulation ppp
!
! The following command defines interface dialer1 as external for NAT.
 ip nat outside
 dialer wait-for-carrier-time 30
 dialer hold-queue 10
 dialer remote-name dallas
 dialer idle-timeout 180
!
! The following command defines the dialer string for the central access server.
 dialer string 4159991234
 dialer pool 1
 dialer-group 1
!
! The following commands define the static route to the WAN interface.
ip route 0.0.0.0 0.0.0.0 dialer1
access-list 101 permit ip 10.0.0.0 0.255.255.255 any
dialer-list 1 protocol ip list 101
```

Virtual Templates, Profiles, and Networks

Configuring Virtual Profiles

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure virtual profiles for use with virtual access interfaces. It includes the following main sections:

Virtual profiles run on all Cisco IOS platforms that support Multilink PPP (MLP).

We recommend that unnumbered addresses be used in virtual template interfaces to ensure that duplicate network addresses are not created on virtual access interfaces.

Virtual profiles interoperate with Cisco dial-on-demand routing (DDR), MLP, and dialers such as ISDN.

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

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

Virtual Profiles Overview

A virtual profile is a unique application that can create and configure a virtual access interface dynamically when a dial-in call is received and that can tear down the interface dynamically when the call ends. Virtual profiles support these encapsulation methods:

- PPP
- MLP
- High-Level Data Link Control (HDLC)
- Link Access Procedure, Balanced (LAPB)
- X.25
- Frame Relay

Any commands for these encapsulations that can be configured under a serial interface can be configured under a virtual profile stored in a user file on an authentication, authorization, and accounting (AAA) server and a virtual profile virtual template configured locally. The AAA server daemon downloads them as text to the network access server and is able to handle multiple download attempts.

The configuration information for a virtual profiles virtual access interface can come from a virtual template interface or from user-specific configuration stored on a AAA server, or both.

If a B interface is bound by the calling line identification (CLID) to a created virtual access interface cloned from a virtual profile or a virtual template interface, only the configuration from the virtual profile or the virtual template takes effect. The configuration on the D interface is ignored unless successful binding occurs by PPP name. Both the link and network protocols run on the virtual access interface instead of the B channel, unless the encapsulation is PPP.

Moreover, in previous releases of Cisco IOS software, downloading a profile from an AAA server and creating and cloning a virtual access interface was always done after the PPP call answer and link control protocol (LCP) up processes. The AAA download is part of authorization. But in the current release, these operations must be performed before the call is answered and the link protocol goes up. This restriction is a new AAA nonauthenticated authorization step. The virtual profile code handles multiple download attempts and identifies whether a virtual access interface was cloned from a downloaded virtual profile.

When a successful download is done through nonauthenticated authorization and the configuration on the virtual profile has encapsulation PPP and PPP authentication, authentication is negotiated as a separate step after LCP comes up.

The per-user configuration feature also uses configuration information gained from a AAA server. However, per-user configuration uses *network* configurations (such as access lists and route filters) downloaded during Network Control Protocol (NCP) negotiations.

Two rules govern virtual access interface configuration by virtual profiles, virtual template interfaces, and AAA configurations:

- Each virtual access application can have at most one template to clone from but can have multiple AAA configurations to clone from (virtual profiles AAA information and AAA per-user configuration, which in turn might include configuration for multiple protocols).
- When virtual profiles are configured by virtual template, its template has higher priority than any other virtual template.

See the section "How Virtual Profiles Work—Four Configuration Cases" for a description of the possible configuration sequences for configuration by virtual template or AAA or both. See the section "Multilink PPP Effect on Virtual Access Interface Configuration" for a description of the possible configuration sequences that depend on the presence or absence by MLP or another virtual access feature that clones a virtual template interface.

DDR Configuration of Physical Interfaces

Virtual profiles fully interoperate with physical interfaces in the following DDR configuration states when no other virtual access interface application is configured:

- Dialer profiles are configured for the interface—The dialer profile is used instead of the virtual profiles configuration.
- DDR is not configured on the interface—Virtual profiles overrides the current configuration.
- Legacy DDR is configured on the interface—Virtual profiles overrides the current configuration.

Note If a dialer interface is used (including any ISDN dialer), its configuration is used on the physical interface instead of the virtual profiles configuration.

Multilink PPP Effect on Virtual Access Interface Configuration

As shown in Table 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—Applies the virtual template.
- Case 2: Virtual Profiles Configured by AAA—Applies the user-specific interface configuration received from the AAA server.
- Case 3: Virtual Profiles Configured by Virtual Template and AAA Configuration—Applies the virtual template and the user-specific interface configuration received from the AAA server.

- Case 4: Virtual Profiles Configured by AAA, and a Virtual Template Defined by Another Application—Applies the other application's virtual template interface and then applies the user-specific interface configuration received from the AAA server.

Note All cases assume that AAA is configured globally on the router, that the user has configuration information in the user file on the AAA server, that PPP authentication and authorization proceed as usual, and that the AAA server sends user-specific configuration information in the authorization approval response packet to the router.

The cases also assume that AAA works as designed and that the AAA server sends configuration information for the dial-in user to the router, even when virtual profiles by virtual template are configured.

See the sections "Virtual Profiles Configured by Virtual Templates," "Virtual Profiles Configured by AAA Configuration," "Virtual Profiles Configured by Virtual Templates and AAA Configuration," and "Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway" later in this chapter for examples of how to configure these cases.

Case 1: Virtual Profiles Configured by Virtual Template

In the case of virtual profiles configured by virtual template, the software functions as follows:

- If the physical interface is configured for dialer profiles (a DDR feature), the router looks for a dialer profile for the specific user.
- If a dialer profile is found, it is used instead of virtual profiles.
- If a dialer profile is not found for the user, or legacy DDR is configured, or DDR is not configured at all, virtual profiles create a virtual access interface for the user.

The router applies the configuration commands that are in the virtual template interface to create and configure the virtual profile. The template includes generic interface information and router-specific information, but no user-specific information. No matter whether a user dialed in on a synchronous serial, an asynchronous serial, or an ISDN interface, the dynamically created virtual profile for the user is configured as specified in the virtual template.

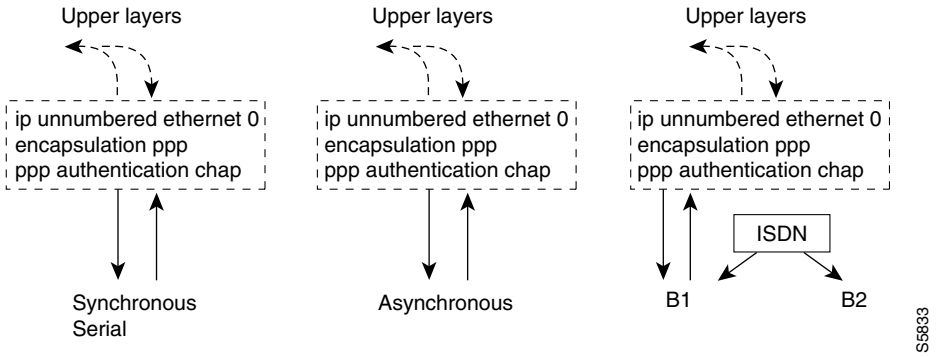
Then the router interprets the lines in the AAA authorization approval response from the server as Cisco IOS commands to apply to the virtual profile for the user.

Data flows through the virtual profile, and the higher layers treat it as the interface for the user.

For example, if a virtual template included only the three commands **ip unnumbered ethernet 0**, **encapsulation ppp,** and **ppp authentication chap**, the virtual profile for any dial-in user would include those three commands.

In Figure 15, 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 15 Virtual Profiles by Virtual Template



See the section "Configuring Virtual Profiles by Virtual Template" later in this chapter for configuration tasks for this case.

Case 2: Virtual Profiles Configured by AAA

In this case, no dialer profile (a DDR feature) is defined for the specific user and no virtual template for virtual profiles is defined, but virtual profiles by AAA are enabled on the router.

During the PPP authorization phase for the user, the AAA server responds as usual to the router. The authorization approval contains configuration information for the user. The router interprets each of the lines in the AAA response from the server as Cisco IOS commands to apply to the virtual profile for the user.

Note If MLP is negotiated, the MLP virtual template is cloned first (this is the second row), and then interface-specific commands included in the AAA response from the server for the user are applied. The MLP virtual template overrides any conflicting interface configuration, and the AAA interface configuration overrides any conflicting configuration from both the physical interface and the MLP virtual template.

The router applies all the user-specific interface commands received from the AAA server.

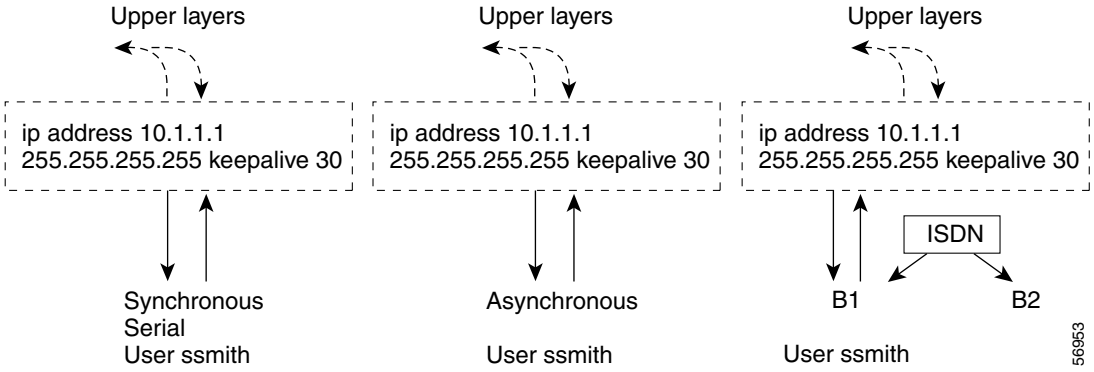
Suppose, for example, that the router interpreted the response by the AAA server as including only the following two commands for this user:

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

In Figure 16, 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 16 Virtual Profiles by AAA Configuration



See the section "Configuring Virtual Profiles by AAA Configuration" later in this chapter for configuration tasks for this case.

Case 3: Virtual Profiles Configured by Virtual Template and AAA Configuration

In this case, no DDR dialer profile is defined for the specific user, a virtual template for virtual profiles is defined, virtual profiles by AAA is enabled on the router, the router is configured for AAA, and a user-specific interface configuration for the user is stored on the AAA server.

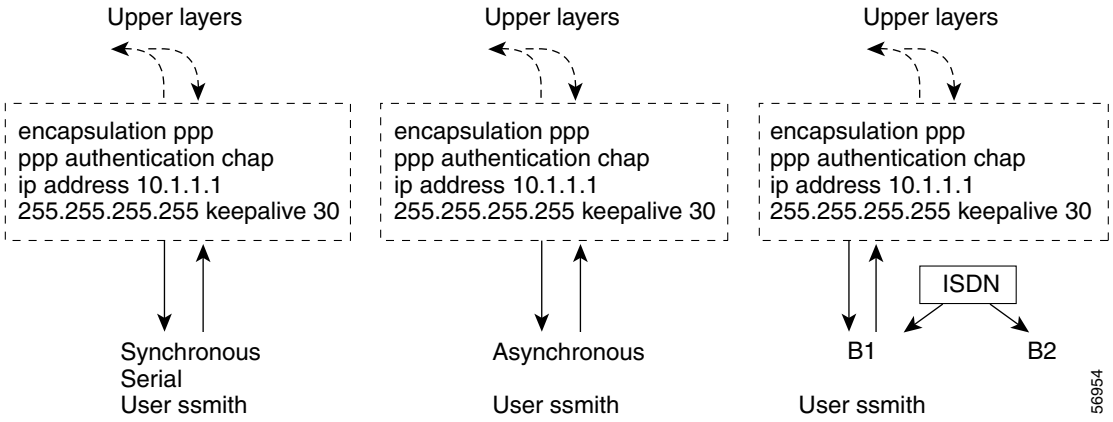
The router performs the following tasks in order:

1. Dynamically creates a virtual access interface cloned from the virtual template defined for virtual profiles.
2. Applies the user-specific interface configuration received from the AAA server.

If any command in the user's configuration conflicts with a command on the original interface or a command applied by cloning the virtual template, the user-specific command overrides the other command.

Suppose that the router had the virtual template as defined in Case 1 and the AAA user configuration as defined in Case 2. In Figure 17 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 17 Virtual Profiles by Both Virtual Template and AAA Configuration



See the section "Configuring Virtual Profiles by Both Virtual Template and AAA Configuration" later in this chapter for configuration tasks for this case.

Case 4: Virtual Profiles Configured by AAA, and a Virtual Template Defined by Another Application

In this case, no DDR dialer profile is defined for the specific user, virtual profiles by AAA are configured on the router but no virtual template is defined for virtual profiles, and a user-specific interface configuration is stored on the AAA server. In addition, a virtual template is configured for some other virtual access application (a VPDN, for example).

The router performs the following tasks in order:

1. Dynamically creates a virtual access interface and clones the virtual template from the other virtual access application onto it.
2. Applies the user-specific interface configuration received from the AAA server.

If any command in the virtual template conflicts with a command on the original interface, the template overrides it.

If any command in the AAA interface configuration for the user conflicts with a command in the virtual template, the user AAA interface configuration conflicts will override the virtual template.

If per-user configuration is also configured on the AAA server, that network protocol configuration is applied to the virtual access interface last.

The result is a virtual interface unique to that user.

How to Configure Virtual Profiles

To configure virtual profiles for dial-in users, perform the tasks in *one* of the first three sections and then troubleshoot the configuration by performing the tasks in the last section:

- Configuring Virtual Profiles by Virtual Template (As required)
- Configuring Virtual Profiles by AAA Configuration (As required)
- Configuring Virtual Profiles by Both Virtual Template and AAA Configuration (As required)

- Troubleshooting Virtual Profile Configurations (As required)

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

See the section "Configuration Examples for Virtual Profiles" at the end of this chapter for examples of how to use virtual profiles in your network configuration.

Configuring Virtual Profiles by Virtual Template

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

- Creating and Configuring a Virtual Template Interface
- Specifying a Virtual Template Interface for Virtual Profiles

Note The order in which these tasks is performed is not crucial. However, both tasks must be completed before virtual profiles are used.

Creating and Configuring a Virtual Template Interface

Because a virtual template interface is a serial interface, all the configuration commands that apply to serial interfaces can also be applied to virtual template interfaces, except **shutdown** and **dialer** commands.

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered ethernet 0`	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template interface.

Other optional PPP configuration commands can be added to the virtual template configuration. For example, you can add the **ppp authentication chap** command.

Specifying a Virtual Template Interface for Virtual Profiles

To specify a virtual template interface as the source of information for virtual profiles, use the following command in global configuration mode:

Command	Purpose
`Router(config)# virtual-profile virtual-template` *number*	Specifies the virtual template interface as the source of information for virtual profiles.

Virtual template numbers range from 1 to 25.

Configuring Virtual Profiles by AAA Configuration

To configure virtual profiles by AAA only, complete these three tasks in any order. All tasks must be completed before virtual profiles are used.

- On the AAA server, create user-specific interface configurations for each of the specific users to use this method. See your AAA server documentation for more detailed configuration information about your AAA server.
- Configure AAA on the router, as described in the *Cisco IOS Security Configuration Guide*, Release 12.2.
- Specify AAA as the source of information for virtual profiles.

To specify AAA as the source of information for virtual profiles, use the following command in global configuration mode:

Command	Purpose
`Router(config)# virtual-profile aaa`	Specifies AAA as the source of user-specific interface configuration.

If you also want to use per-user configuration for network protocol access lists or route filters for individual users, see the chapter "Configuring Per-User Configuration" in this publication. In this case, no virtual template interface is defined for virtual profiles.

Configuring Virtual Profiles by Both Virtual Template and AAA Configuration

Use of user-specific AAA interface configuration information with virtual profiles requires the router to be configured for AAA and requires the AAA server to have user-specific interface configuration AV-pairs. The relevant AV-pairs (on a RADIUS server) begin as follows:

```
cisco-avpair = "lcp:interface-config=...",
```

The information that follows the equal sign (=) could be any Cisco IOS interface configuration command. For example, the line might be the following:

```
cisco-avpair = "lcp:interface-config=ip address 192.168.200.200 255.255.255.0",
```

Use of a virtual template interface with virtual profiles requires a virtual template to be defined specifically for virtual profiles.

To configure virtual profiles by both virtual template interface and AAA configuration, complete the following tasks in any order. All tasks must be completed before virtual profiles are used.

- On the AAA server, create user-specific interface configurations for each of the specific users to use this method. See your AAA server documentation for more detailed configuration information about your AAA server.
- Configure AAA on the router, as described in the *Cisco IOS Security Configuration Guide* publication.
- Creating and Configuring a Virtual Template Interface, described later in this chapter.
- Specifying Virtual Profiles by Both Virtual Templates and AAA, described later in this chapter.

Creating and Configuring a Virtual Template Interface

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template` *number*	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered ethernet 0`	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template interface.

Because the software treats a virtual template interface as a serial interface, all the configuration commands that apply to serial interfaces can also be applied to virtual template interfaces, except **shutdown** and **dialer** commands. Other optional PPP configuration commands can also be added to the virtual template configuration. For example, you can add the **ppp authentication chap** command.

Specifying Virtual Profiles by Both Virtual Templates and AAA

To specify both the virtual template interface and the AAA per-user configuration as sources of information for virtual profiles, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# virtual-profile virtual-template` *number*	Defines the virtual template interface as the source of information for virtual profiles.
Step 2	`Router(config)# virtual-profile aaa`	Specifies AAA as the source of user-specific configuration for virtual profiles.

If you also want to use per-user configuration for network protocol access lists or route filters for individual users, see the chapter “Configuring Per-User Configuration” in this publication.

Troubleshooting Virtual Profile Configurations

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

Command	Purpose
`Router# debug dialer`	Displays information about dial calls and negotiations and virtual profile events.
`Router# debug aaa per-user`	Displays information about the per-user configuration downloaded from the AAA server.
`Router# debug vtemplate`	Displays cloning information for a virtual access interface from the time it is cloned from a virtual template to the time it comes down.

Configuration Examples for Virtual Profiles

The following sections provide examples for the four cases described in this chapter:

- Virtual Profiles Configured by Virtual Templates
- Virtual Profiles Configured by AAA Configuration
- Virtual Profiles Configured by Virtual Templates and AAA Configuration
- Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway

In these examples, BRI 0 is configured for legacy DDR, and interface BRI 1 is configured for dialer profiles. Note that interface dialer 0 is configured for legacy DDR. Interface dialer 1 is a dialer profile.

The intention of the examples is to show how to configure virtual profiles. In addition, the examples show the interoperability of DDR and dialer profiles in the respective cases with various forms of virtual profiles.

The same user names (John and Rick) occur in all these examples. Note the different configuration allowed to them in each of the four examples.

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

To see an example of a configuration using virtual profiles and the Dynamic Multiple Encapsulations feature, see the "Multiple Encapsulations over ISDN" example in the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles."

Virtual Profiles Configured by Virtual Templates

The following example shows a router configured for virtual profiles by virtual template. (Virtual profiles do not have any interface-specific AAA configuration.) Comments in the example draw attention to specific features or ignored lines.

In this example, the same virtual template interface applies to both users; they have the same interface configurations.

Router Configuration

```
! Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
! The following command is required.
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
! Specify configuration of virtual profiles by virtual template.
! This is the key command for this example.
virtual-profile virtual-template 1
!
! Define the virtual template.
interface Virtual-Template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp authentication chap
!
switch-type basic-dms100
```

```
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
interface BRI 1
 description Connected to 104
 encapsulation ppp
! Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR for John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
! Enable legacy DDR.
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 1 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
! Disable fast switching.
 no ip route-cache
 ppp authentication chap
 dialer-list 1 protocol ip permit
```

Virtual Profiles Configured by AAA Configuration

The following example shows the router configuration for virtual profiles by AAA and the AAA server configuration for user-specific interface configurations. John and Rick have different IP addresses.

In the AAA configuration cisco-avpair lines, "\n" is used to indicate the start of a new Cisco IOS command line.

AAA Configuration for John and Rick

```
john Password = "welcome"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 75\nip address 172.16.100.100
 255.255.255.0",
rick Password = "emoclew"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 100\nip address 192.168.200.200
 255.255.255.0"
```

Router Configuration

```
! Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
! This is a key command for this example.
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
! Specify configuration of virtual profiles by aaa.
! This is a key command for this example.
virtual-profiles aaa
!
! Interface BRI 0 is configured for legacy DDR.
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
! Interface BRI 1 is configured for dialer profiles.
interface BRI 1
 description Connected to 104
 encapsulation ppp
! Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR for John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
! Enable legacy DDR.
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 1 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
! Disable fast switching.
 no ip route-cache
 ppp authentication chap
 dialer-list 1 protocol ip permit
```

Virtual Profiles Configured by Virtual Templates and AAA Configuration

The following example shows how virtual profiles can be configured by both virtual templates and AAA configuration. John and Rick can dial in from anywhere and have their same keepalive settings and their own IP addresses.

The remaining AV pair settings are not used by virtual profiles. They are the network protocol access lists and route filters used by AAA-based per-user configuration.

In the AAA configuration cisco-avpair lines, "\n" is used to indicate the start of a new Cisco IOS command line.

AAA Configuration for John and Rick

```
john Password = "welcome"
     User-Service-Type = Framed-User,
     Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=keepalive 75\nip address 10.16.100.100
255.255.255.0",
        cisco-avpair = "ip:rte-fltr-out#0=router igrp 60",
        cisco-avpair = "ip:rte-fltr-out#3=deny 172.16.0.0 0.255.255.255",
        cisco-avpair = "ip:rte-fltr-out#4=deny 172.17.0.0 0.255.255.255",
        cisco-avpair = "ip:rte-fltr-out#5=permit any"
rick Password = "emoclew"
     User-Service-Type = Framed-User,
     Framed-Protocol = PPP,
        cisco-avpair = "lcp:interface-config=keepalive 100\nip address 192.168.200.200
255.255.255.0",
        cisco-avpair = "ip:inacl#3=permit ip any any precedence immediate",
        cisco-avpair = "ip:inacl#4=deny igrp 10.0.1.2 255.255.0.0 any",
        cisco-avpair = "ip:outacl#2=permit ip any any precedence immediate",
        cisco-avpair = "ip:outacl#3=deny igrp 10.0.9.10 255.255.0.0 any"
```

Router Configuration

```
! Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
! This is a key command for this example.
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
! Specify use of virtual profiles and a virtual template.
! The following two commands are key for this example.
virtual-profile virtual-template 1
virtual-profile aaa
!
! Define the virtual template.
interface Virtual-Template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp authentication chap
!
! Interface BRI 0 is configured for legacy DDR.
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
! Interface BRI 1 is configured for dialer profiles.
```

```
interface BRI 1
 description Connected to 104
 encapsulation ppp
! Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR to dial out to John and Rick.
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
! Configure dialer interface 0 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
! Disable fast switching.
 no ip route-cache
 ppp authentication chap
!
 dialer-list 1 protocol ip permit
```

Virtual Profiles Configured by AAA Plus a VPDN Virtual Template on a VPDN Home Gateway

Like the virtual profiles configured by AAA example earlier in this section, the following example shows the router configuration for virtual profiles by AAA. The user file on the AAA server also includes interface configuration for John and Rick, the two users. Specifically, John and Rick each have their own IP addresses when they are in privileged mode.

In this case, however, the router is also configured as the VPDN home gateway. It clones the VPDN virtual template interface first and then clones the virtual profiles AAA interface configuration. If per-user configuration were configured on this router and the user file on the AAA server had network protocol information for the two users, that information would be applied to the virtual access interface last.

In the AAA configuration cisco-avpair lines, “\n” is used to indicate the start of a new Cisco IOS command line.

AAA Configuration for John and Rick

```
john Password = "welcome"
 User-Service-Type = Framed-User,
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 75\nip address 10.100.100.100
 255.255.255.0",
rick Password = "emoclew"
 User-Service-Type = Framed-User,
```

```
 Framed-Protocol = PPP,
 cisco-avpair = "lcp:interface-config=keepalive 100\nip address 192.168.200.200
 255.255.255.0"
```

Router Configuration

```
!Configure the router as the VPDN home gateway.
!
!Enable VPDN and specify the VPDN virtual template to use on incoming calls from the
!network access server.
vpdn enable
vpdn incoming dallas_wan go_blue virtual-template 6
!
!Configure the virtual template interface for VPDN.
interface virtual template 6
ip unnumbered ethernet 0
encapsulation ppp
ppp authentication chap
!
!Enable AAA on the router.
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
!Specify configuration of virtual profiles by aaa.
virtual-profiles aaa
!
!Configure the physical synchronous serial 0 interface.
interface Serial 0
 description Connected to 101
 encapsulation ppp
!Disable fast switching.
 no ip route-cache
 ppp authentication chap
!
!Configure serial interface 1 for DDR. S1 uses dialer rotary group 0, which is
!defined on BRI interface 0.
interface serial 1
 description Connected to 102
 encapsulation ppp
 dialer in-band
! Disable fast switching.
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
interface BRI 0
 description Connected to 103
 encapsulation ppp
 no ip route-cache
 dialer rotary-group 0
 ppp authentication chap
!
interface BRI 1
 description Connected to 104
 encapsulation ppp
!Disable fast switching.
 no ip route-cache
 dialer pool-member 1
 ppp authentication chap
!
!Configure dialer interface 0 for DDR to call and receive calls from John and Rick.
```

```
interface dialer 0
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
!Enable legacy DDR.
 dialer in-band
!Disable fast switching.
 no ip route-cache
 dialer map ip 10.1.1.2 name john 1111
 dialer map ip 10.1.1.3 name rick 2222
 dialer-group 1
 ppp authentication chap
!
!Configure dialer interface 1 for DDR to dial out to Rick.
interface dialer 1
 ip address 10.2.2.2 255.255.255.0
 encapsulation ppp
 dialer remote-name rick
 dialer string 3333
 dialer pool 1
 dialer-group 1
!Disable fast switching.
 no ip route-cache
 ppp authentication chap
 dialer-list 1 protocol ip permit
```

Configuring Virtual Template Interfaces

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure virtual template interfaces. It includes the following main sections:

The following template and virtual interface limitations apply:

- Although a system can generally support many virtual template interfaces, one template for each virtual access application is a more realistic limit.
- When in use, each virtual access interface cloned from a template requires the same amount of memory as a serial interface. Limits to the number of virtual access interfaces that can be configured are determined by the platform.
- Virtual access interfaces are not directly configurable by users, except by configuring a virtual template interface or including the configuration information of the user (through virtual profiles or per-user configuration) on an authentication, authorization, and accounting (AAA) server. However, information about an in-use virtual access interface can be displayed, and the virtual access interface can be cleared.
- Virtual interface templates provide no *direct* value to users; they must be applied to or associated with a virtual access feature using a command with the **virtual-template** keyword.

  For example, the **interface virtual-template** command creates the virtual template interface and the **multilink virtual-template** command applies the virtual template to a multilink stack group. The **virtual-profile virtual-template** command specifies that a virtual template interface will be used as a source of configuration information for virtual profiles.

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

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

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

Virtual Template Interface Service Overview

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

Virtual template interfaces can be configured independently of any physical interface and applied dynamically, as needed, to create virtual access interfaces. When a user dials in, a predefined configuration template is used to configure a virtual access interface; when the user is done, the virtual access interface goes down and the resources are freed for other dial-in uses.

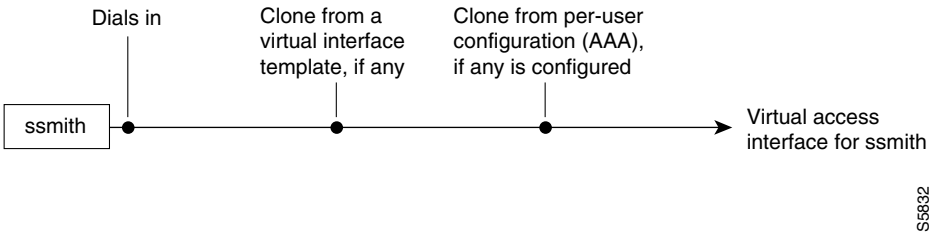
A virtual template interface is a logical entity—a configuration for a serial interface but not tied to a physical interface—that can be applied dynamically as needed. Virtual access interfaces are virtual interfaces that are created, configured dynamically (for example, by *cloning* a virtual template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

Virtual template interfaces are one possible source of configuration information for a virtual access interface.

Each virtual access interface can clone from only one template. But some applications can take configuration information from multiple sources; for example, virtual profiles can take configuration information from a virtual template interface, or from interface-specific configuration information stored from a user on a AAA server, or from network protocol configuration from a user stored on a AAA server, or all three. The result of using template and AAA configuration sources is a virtual access interface uniquely configured for a specific dial-in user.

Figure 14 illustrates that a router can create a virtual access interface by first using the information from a virtual template interface (if any is defined for the application) and then using the information in a per-user configuration (if AAA is configured on the router and virtual profiles or per-user configuration or both are defined for the specific user).

Figure 14 Possible Configuration Sources for Virtual Access Interfaces



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

- For easier maintenance, allows customized configurations to be predefined and then applied dynamically when the specific need arises.

- For scalability, allows interface configuration to be separated from physical interfaces. Virtual interfaces can share characteristics, no matter what specific type of interface the user called on.
- For consistency and configuration ease, allows the same predefined template to be used for all users dialing in for a specific application.
- For efficient router operation, frees the virtual access interface memory for another dial-in use when the call from the user ends.

Features that Apply Virtual Template Interfaces

The following features apply virtual template interfaces to create virtual access interfaces dynamically:

- Virtual profiles
- Virtual Private Dialup Networks (VPDN)
- Multilink PPP (MLP)
- Multichassis Multilink PPP (MMP)
- Virtual templates for protocol translation
- PPP over ATM

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

To create and configure a virtual template interface, compete the tasks in this chapter. To apply a virtual template interface, refer to the specific feature that applies the virtual template interface.

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

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

Selective Virtual Access Interface Creation

Optionally, you can configure a router to automatically determine whether to create a virtual access interface for each inbound connection. In particular, a call that is received on a physical asynchronous interface that uses a AAA per-user configuration can now be processed without a virtual access interface being created by a router that is also configured for virtual profiles.

The following three criteria determine whether a virtual access interface is created:

- Is there a virtual profile AAA configuration?
- Is there a AAA per-user configuration?
- Does the link interface support direct per-user AAA?

A virtual access interface *will* be created in the following scenarios:

- If there *is* a virtual profile AAA configuration.
- If there *is not* a virtual profile AAA configuration, but there *is* a AAA per-user configuration *and* the link interface *does not* support direct per-user AAA (such as ISDN).

A virtual access interface *will not* be created in the following scenarios:

- If there is *neither* a virtual profile AAA configuration *nor* a AAA per-user configuration.

- If there is *not* a virtual profile AAA configuration, but there *is* a AAA per-user configuration and the link interface does support direct per-user AAA (such as asynchronous).

How to Configure a Virtual Template Interface

To create and configure a virtual template interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface virtual-template number`	Creates a virtual template interface and enters interface configuration mode.
Step 2	`Router(config-if)# ip unnumbered ethernet 0`	Enables IP without assigning a specific IP address on the LAN.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the virtual template Interface.
Step 4	`Router(config-if)# virtual-profile if-needed`	(Optional) Creates virtual-access interfaces only if the inbound connection requires one.

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

Optionally, other PPP configuration commands can be added to the virtual template configuration. For example, you can add the **ppp authentication chap** command.

All configuration commands that apply to serial interfaces can also be applied to virtual template interfaces, except **shutdown** and **dialer** commands.

For virtual template interface examples, see the "Configuration Examples for Virtual Template Interface" section later in this chapter.

Monitoring and Maintaining a Virtual Access Interface

When a virtual template interface or a configuration from a user on a AAA server or both are applied dynamically, a virtual access interface is created. Although a virtual access interface cannot be created and configured directly, it can be displayed and cleared.

To display or clear a specific virtual access interface, use the following commands in EXEC mode:

Command	Purpose
`Router> show interfaces virtual-access number`	Displays the configuration of the virtual access interface.
`Router> clear interface virtual-access number`	Tears down the virtual access interface and frees the memory for other dial-in uses.

Configuration Examples for Virtual Template Interface

The following sections provide virtual template interface configuration examples:

- Basic PPP Virtual Template Interface
- Virtual Template Interface

- Selective Virtual Access Interface
- RADIUS Per-User and Virtual Profiles
- TACACS+ Per-User and Virtual Profiles

Basic PPP Virtual Template Interface

The following example enables virtual profiles (configured only by virtual template) on straightforward PPP (no MLP), and configures a virtual template interface that can be cloned on a virtual access interface for dial-in users:

```
virtual-profile virtual-template 1

interface virtual-template 1
 ip unnumbered ethernet 0
 encapsulation ppp
 ppp authentication chap
```

Virtual Template Interface

The following two examples configure a virtual template interface and then display the configuration of a virtual access interface when the template interface has been applied.

This example uses a named Internet Protocol Exchange (IPX) access list:

```
Router(config)# interface virtual-template 1
 ip unnumbered Ethernet0
 ipx ppp-client Loopback2
 no cdp enable
 ppp authentication chap
```

This example displays the configuration of the active virtual access interface that was configured by virtual-template 1, defined in the preceding example:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
Router# show interfaces virtual-access 1 configuration

Virtual-Access1 is a L2F link interface
interface Virtual-Access1 configuration...
ip unnumbered Ethernet0
ipx ppp-client Loopback2
no cdp enable
ppp authentication chap
```

Selective Virtual Access Interface

The following example shows how to create a virtual access interface for incoming calls that require a virtual access interface:

```
aaa new-model
aaa authentication ppp default local radius tacacs
aaa authorization network default local radius tacacs
```

```
virtual-profile if-needed
virtual-profile virtual-template 1
virtual-profile aaa
!
interface Virtual-Template1
 ip unnumbered Ethernet 0
 no ip directed-broadcast
 no keepalive
 ppp authentication chap
 ppp multilink
```

RADIUS Per-User and Virtual Profiles

The following examples show RADIUS user profiles that could be used for selective virtual access interface creation.

This example shows AAA per-user configuration for a RADIUS user profile:

```
RADIUS user profile:
        foo     Password = "test"
                 User-Service-Type = Framed-User,
                 Framed-Protocol = PPP,
                cisco-avpair = "ip:inacl#1=deny 10.10.10.10 0.0.0.0",
                cisco-avpair = "ip:inacl#1=permit any"
```

This example shows a virtual profile AAA configuration for a RADIUS user profile:

```
RADIUS user profile:
       foo  Password = "test"
            User-Service-Type = Framed-User,
            Framed-Protocol = PPP,
          cisco-avpair = "lcp:interface-config=keepalive 30\nppp max-bad-auth 4"
```

TACACS+ Per-User and Virtual Profiles

The following examples show TACACS+ user profiles that could be used for selective virtual access interface creation.

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

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

This example shows a virtual profile AAA configuration for a TACACS+ user profile:

```
TACACS+ user profile:
        user = foo {
                name = "foo"
                global = cleartext test
                service = PPP protocol= lcp {
                        interface-config="keepalive 30\nppp max-bad-auth 4"
                }
                service = ppp protocol = ip {
                }
```

```
}
```

PPP Configuration

Configuring Asynchronous SLIP and PPP

Configuring Asynchronous SLIP and PPP

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure asynchronous Serial Line Internet Protocol (SLIP) and PPP. It includes the following main sections:

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

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

Asynchronous SLIP and PPP Overview

PPP and SLIP define methods of sending IP packets over standard asynchronous serial lines with minimum line speeds of 1200 baud.

Using SLIP or PPP encapsulation over asynchronous lines is an inexpensive way to connect personal computers (PCs) to a network. PPP and SLIP over asynchronous dialup modems allow a home computer to be connected to a network without the cost of a leased line. Dialup PPP and SLIP links can also be used for remote sites that need only occasional remote node or backup connectivity. Both public-domain and vendor-supported PPP and SLIP implementations are available for a variety of computer applications.

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

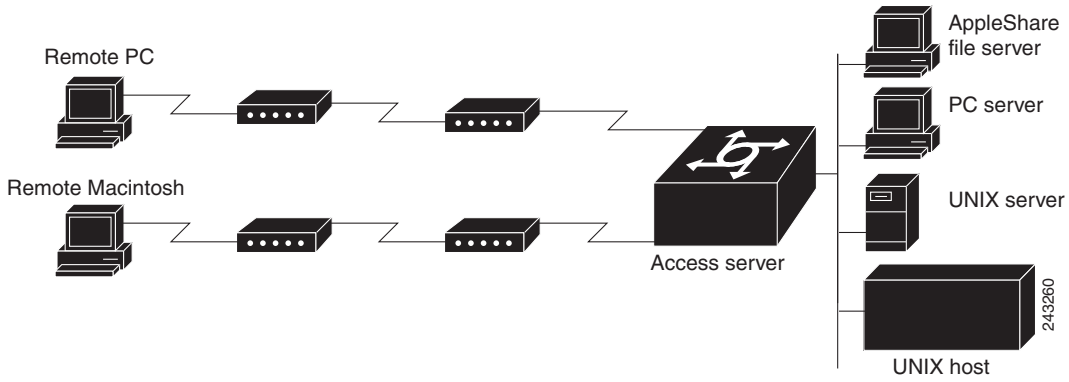
The Cisco IOS software concentrates a large number of SLIP or PPP PC or workstation client hosts onto a network interface that allows the PCs to communicate with any host on the network. The Cisco IOS software can support any combination of SLIP or PPP lines and lines dedicated to normal asynchronous devices such as terminals and modems. Refer to RFC 1055 for more information about SLIP, and RFCs 1331 and 1332 for more information about PPP.

SLIP is an older protocol. PPP is a newer, more robust protocol than SLIP, and it contains functions that can detect or prevent misconfiguration. PPP also provides greater built-in security mechanisms.

Note Most asynchronous serial links have very low bandwidth. Take care to configure your system so the links will not be overloaded. Consider using default routes and filtering routing updates to prevent them from being sent on these asynchronous lines.

Figure 18 illustrates a typical asynchronous SLIP or PPP remote-node configuration.

Figure 18 Sample SLIP or PPP Remote-Node Configuration



Responding to BOOTP Requests

The BOOTP protocol allows a client machine to discover its own IP address, the address of the router, and the name of a file to be loaded in to memory and executed. There are typically two phases to using BOOTP: first, the client's address is determined and the boot file is selected; then the file is transferred, typically using the TFTP.

PPP and SLIP clients can send BOOTP requests to the Cisco IOS software, and the Cisco IOS software responds with information about the network. For example, the client can send a BOOTP request to learn its IP address and where the boot file is located, and the Cisco IOS software responds with the information.

BOOTP supports the extended BOOTP requests specified in RFC 1084 and works for both PPP and SLIP encapsulation.

BOOTP compares to Reverse Address Resolution Protocol (RARP) as follows: RARP is an older protocol that allows a client to determine its IP address if it knows its hardware address. (Refer to the *Cisco IOS IP Configuration Guide* for more information about RARP.) However, RARP is a hardware link protocol, so it can be implemented only on hosts that have special kernel or driver modifications that allow access to these raw packets. BOOTP does not require kernel modifications.

Asynchronous Network Connections and Routing

Line configuration commands configure a connection to a terminal or a modem. Interface configuration (**async**) commands, described in this chapter, configure a line as an asynchronous network interface over which networking functions are performed.

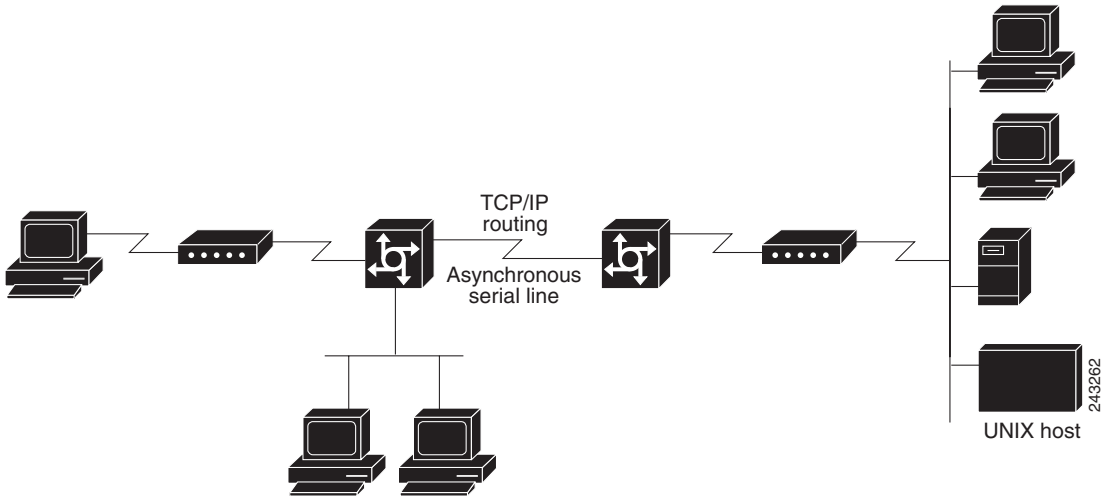
The Cisco IOS software also supports IP routing connections for communication that requires connecting one network to another.

The Cisco IOS software supports protocol translation for PPP and SLIP between other network devices running Telnet, local-area transport (LAT), or X.25. For example, you can send IP packets across a public X.25 packet assembler/disassembler (PAD) network using SLIP or PPP encapsulation when SLIP or PPP protocol translation is enabled. For more information, see the chapter "Configuring Protocol Translation and Virtual Asynchronous Devices" in this publication.

If asynchronous dynamic routing is enabled, you can enable routing at the user level by using the **routing** keyword with the **slip** or **ppp** EXEC command.

Asynchronous interfaces offer both dedicated and dynamic address assignment, configurable hold queues and IP packet sizes, extended BOOTP requests, and permit and deny conditions for controlling access to lines. Figure 19 shows a sample asynchronous routing configuration.

Figure 19 Sample Asynchronous Routing Configuration



SAsynchronous Interfaces and Broadcasts

The Cisco IOS software recognizes a variety of IP broadcast addresses. When a router receives an IP packet from an asynchronous client, it rebroadcasts the packet onto the network without changing the IP header.

The Cisco IOS software receives the SLIP or PPP client broadcasts and responds to BOOTP requests with the current IP address assigned to the asynchronous interface from which the request was received. This facility allows the asynchronous client software to automatically learn its own IP address.

How to Configure Asynchronous SLIP and PPP

To configure SLIP and PPP, perform the tasks in the following sections; all tasks are optional:

- Configuring Network-Layer Protocols over PPP and SLIP (Optional)
- Configuring Asynchronous Host Mobility (Optional)
- Making Additional Remote Node Connections (Optional)
- Configuring Remote Access to NetBEUI Services (Optional)
- Configuring Performance Parameters (Optional)

Configuring Network-Layer Protocols over PPP and SLIP

You can configure network-layer protocols, such as AppleTalk, IP, and Internet Protocol Exchange (IPX), over PPP and SLIP. SLIP supports only IP, but PPP supports each of these protocols. See the sections that follow to configure these protocols over PPP and SLIP.

Configuring IP and PPP

To enable IP-PPP (IPCP) on a synchronous or asynchronous interface, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip address` *ip-address mask* `[secondary]`	Configures IP routing on the interface.
	or	
	`Router(config-if)# ip unnumbered` *type number*	Configures IP unnumbered routing on a serial interface.
Step 2	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the serial interface.
Step 3	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.

Configuring IPX and PPP

You can configure IPX over PPP (IPXCP) on synchronous serial and asynchronous serial interfaces using one of two methods.

The first method associates an asynchronous interface with a loopback interface configured to run IPX. It permits you to configure IPX-PPP on asynchronous interfaces only.

The second method permits you to configure IPX-PPP on asynchronous and synchronous serial interfaces. However, it requires that you specify a dedicated IPX network number for each interface, which can require a substantial number of network numbers for a large number of interfaces.

You can also configure IPX to run on virtual terminal lines configured for PPP. See the section "Enabling IPX and PPP over X.25 to an IPX Network on Virtual Terminal Lines" later in this chapter.

Note If you are configuring IPX-PPP on asynchronous interfaces, you should filter routing updates on the interface. Most asynchronous serial links have very low bandwidth, and routing updates take up a great deal of bandwidth. The previous task table uses the **ipx update interval** command to filter SAP updates.

For more information about filtering routing updates, see the section about creating filters for updating the routing table in the chapter "Configuring Novell IPX" in the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

IPX and PPP and Associating Asynchronous Interfaces with Loopback Interfaces

To permit IPX client connections to an asynchronous interface, the interface must be associated with a loopback interface configured to run IPX. To permit such connections, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx routing [node]`	Enables IPX routing.
Step 2	`Router(config)# interface loopback number`	Creates a loopback interface, which is a virtual interface existing only inside the router, and begins interface configuration mode.
Step 3	`Router(config-if)# ipx network network`[1]	Enables IPX routing on the loopback interface.
Step 4	`Router(config-if)# exit`	Exits to global configuration mode.
Step 5	`Router(config)# interface async number`	Enters interface configuration mode for the asynchronous interface.
Step 6	`Router(config-if)# ip unnumbered type number`	Configures IP unnumbered routing on the interface.
Step 7	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 8	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.
Step 9	`Router(config-if)# ipx ppp-client loopback number`	Assigns the asynchronous interface to the loopback interface configured for IPX.
Step 10	`Router(config-if)# ipx update interval`	Turns off Service Advertising Protocol (SAP) updates to optimize bandwidth on asynchronous interfaces.

1. Every interface must have a unique IPX network number.

IPX and PPP Using Dedicated IPX Network Numbers for Each Interface

To enable IPX and PPP, use the following commands beginning in global configuration mode. The first five steps are required. The last step is optional.

	Command	Purpose
Step 1	`Router(config)# ipx routing [node]`	Enables IPX routing.
Step 2	`Router(config)# interface loopback number`	Creates a loopback interface, which is a virtual interface existing only inside the router, and begins interface configuration mode.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on the interface.
Step 4	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.
Step 5	`Router(config-if)# ipx network network`[1]	Enables IPX routing on the interface.
Step 6	`Router(config-if)# ipx update interval`	(Optional) Turns off SAP updates to optimize bandwidth on asynchronous interfaces.

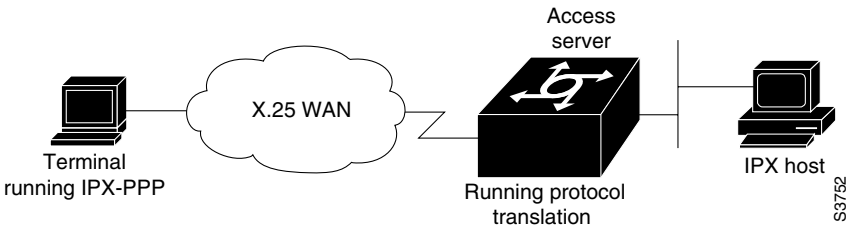
1. Every interface must have a unique IPX network number.

Enabling IPX and PPP over X.25 to an IPX Network on Virtual Terminal Lines

You can enable IPX-PPP on virtual terminal lines, which permits clients to log in to a virtual terminal on a router, invoke a PPP session at the EXEC prompt to a host, and run IPX to the host.

For example, in Figure 20, the client terminal on the X.25 network logs in to the access server via a virtual terminal line, which is configured for IPX-PPP. When the user connects to the access server and the EXEC prompt appears, enter the PPP command to connect to the IPX host. The virtual terminal is configured to run IPX, so when the PPP session is established from the access server, the terminal can access the IPX host using an IPX application.

Figure 20 IPX-PPP on a Virtual Asynchronous Interface



To enable IPX to run over your PPP sessions on virtual terminal lines, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ipx routing [node]`	Enables IPX routing.
Step 2	`Router(config)# interface loopback number`	Creates a loopback interface and begins interface configuration mode.
Step 3	`Router(config-if)# ipx network network`[1]	Enables a virtual IPX network on the loopback interface.
Step 4	`Router(config-if)# vty-async ipx ppp-client loopback number`	Enables IPX-PPP on virtual terminal lines by assigning it to the loopback interface configured for IPX.

1. Every loopback interface must have a unique IPX network number.

Configuring AppleTalk and PPP

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

You create a virtual network that exists only for accessing an AppleTalk internet through the server. To create a new AppleTalk zone, enter the **appletalk virtual-net** command and use a new zone name; this network number is then the only one associated with this zone. To add network numbers to an existing AppleTalk zone, use this existing zone name in the command; this network number is then added to the existing zone. Routing is not supported on these interfaces.

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

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Defines encapsulation as PPP on this interface.
Step 2	`Router(config-if)# appletalk virtual-net network-number zone-name`	Creates an internal network on the server.
Step 3	`Router(config-if)# appletalk client-mode`	Enables client-mode on this interface.

Configuring IP and SLIP

To enable IP-SLIP on a synchronous or asynchronous interface, use the following commands in interface configuration mode:

	Command	Purpose

Step 1	`Router(config-if)# ip address ip-address mask`
	or
	`Router(config-if)# ip unnumbered type number`
		Configures IP routing on the interface.
		
		Configures IP unnumbered routing on a serial interface.

Step 2	`Router(config-if)# encapsulation slip`	Enables SLIP encapsulation on the serial interface.
Step 3	`Router(config-if)# async mode interactive`	Enables interactive mode on an asynchronous interface.

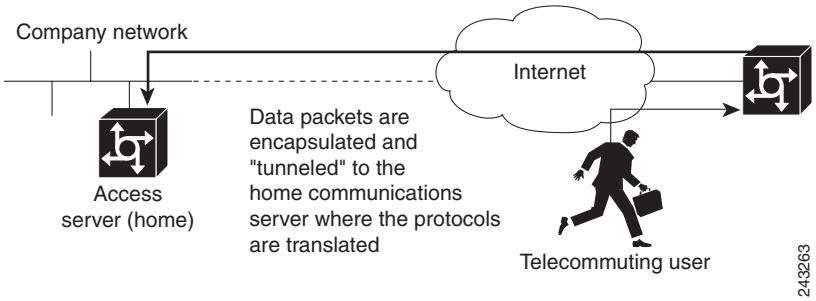
Configuring Asynchronous Host Mobility

The access server supports a packet tunneling strategy that extends the internetwork—in effect creating a virtual private link for the mobile user. When a user activates asynchronous host mobility, the access server on which the remote user dials in becomes a remote point of presence (POP) for the home network of the user. Once logged in, users experience a server environment identical to the one that they experience when they connect directly to the “home” access server.

Once the network-layer connection is made, data packets are tunneled at the physical or data link layer instead of at the protocol layer. In this way, raw data bytes from dial-in users are transported directly to the “home” access server, which processes the protocols.

Figure 21 illustrates the implementation of asynchronous host mobility on an extended internetwork. A mobile user connects to an access server on the internetwork and, by activating asynchronous host mobility, is connected to a “home” access server configured with the appropriate username. The user sees an authentication dialog or prompt from the “home” system and can proceed as if he or she were connected directly to that device.

Figure 21 Asynchronous Host Mobility



Asynchronous host mobility is enabled with the **tunnel** EXEC command and the **ip tcp async-mobility server** global configuration command. The **ip tcp async-mobility server** command establishes asynchronous listening on TCP tunnel port 57. The **tunnel** command sets up a network-layer connection to the specified destination. Both commands must be used. The access server accepts the connection, attaches it to a virtual terminal line, and runs a command parser capable of running the normal dial-in services. After the connection is established, data is transferred between the modem and network connection with a minimum of interpretations. When communications are complete, the network connection can be closed and terminated from either end.

To enable asynchronous host mobility, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip tcp async-mobility server`	Enables asynchronous listening on TCP tunnel port 57.
Step 2	`Router(config)# exit`	Returns to user EXEC mode.
Step 3	`Router# tunnel host`	Sets up a network-layer connection to a router by specifying its Internet name or address. Replace the *host* argument with the name or address of the device that you want to connect to.

To connect from a router other than a Cisco router, you must use Telnet. After a connection is established, you receive an authentication dialog or prompt from your home router, and can proceed as if you are connected directly to that router. When communications are complete, the network connection can be closed and terminated from either end of the connection.

Making Additional Remote Node Connections

This section describes how to connect devices across telephone lines by using PPP and SLIP. It includes the following sections:

- Creating PPP Connections
- Making SLIP Connections

Creating PPP Connections

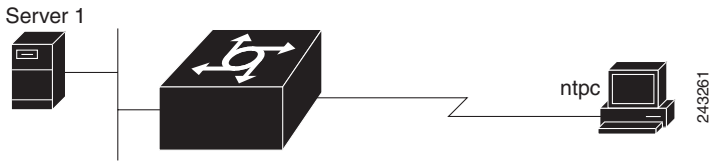
When you connect from a remote node computer through an asynchronous port on an access server to the EXEC facility to connect from the access server to a device on the network, use the following command in EXEC mode:

Command	Purpose
Router> **ppp** {**/default** | {*remote-ip-address* | *remote-name*} [**@***tacacs-server*]} [**/routing**]	Creates a PPP connection.

If you specify an address for the TACACS server using **/default** or *tacacs-server*, the address must be the first parameter in the command after you type **ppp**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

For example, if you are working at home on the device named ntpc in Figure 22 and want to connect to Server 1 using PPP, you could dial in to the access server. When you connect to the EXEC prompt on the access server, enter the **ppp** command to connect with the device.

Figure 22 Using the ppp Command



To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from EXEC mode by using the **exit** command.

Making SLIP Connections

To make a serial connection to a remote host by using SLIP, use the following command in EXEC mode:

Command	Purpose
Router> **slip** [**/default**] {*remote-ip-address* | *remote-name*} [**@***tacacs-server*] [**/routing**]} [**/compressed**]	Creates a SLIP connection.

Your system administrator can configure SLIP to expect a specific address or to provide one for you. It is also possible to set up SLIP in a mode that compresses packets for more efficient use of bandwidth on the line.

If you specify an address for the TACACS server using **/default** or *tacacs-server*, the address must be the first parameter in the command after you type **slip**. If you do not specify an address or enter **/default**, you are prompted for an IP address or host name. You can enter **/default** at this point.

If you do not use the *tacacs-server* argument to specify a TACACS server for SLIP address authentication, the TACACS server specified at login (if any) is used for the SLIP address query.

To optimize bandwidth on a line, SLIP enables compression of the SLIP packets using Van Jacobson TCP header compression as defined in RFC 1144.

To terminate a session, disconnect from the device on the network using the command specific to that device. Then, exit from EXEC mode by using the **exit** command.

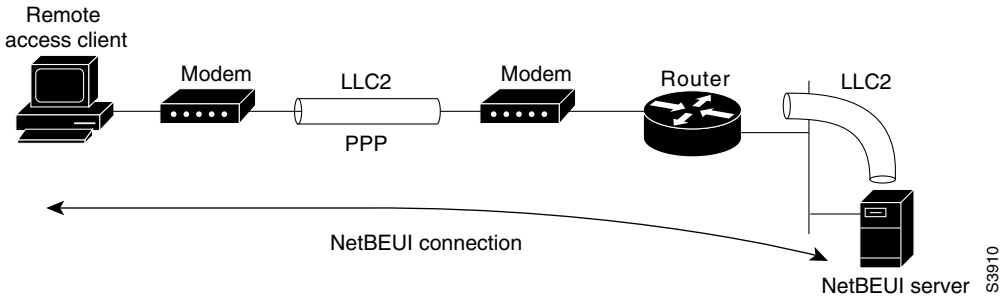
Configuring Remote Access to NetBEUI Services

NetBIOS Extended User Interface (NetBEUI) is a simple networking protocol developed by IBM for use by PCs in a LAN environment. It is an extension of the original Network Basic Input/Output System (NetBIOS) from IBM. NetBEUI uses a broadcast-based name to 802.x address translation mechanism. Because NetBEUI has no network layer, it is a nonroutable protocol.

The NetBIOS Frames Control Protocol (NBFCP) enables packets from a NetBEUI application to be transferred via a PPP connection. NetBEUI/PPP is supported in the access server and Cisco enterprise images only.

Using the Cisco IOS implementation, remote NetBEUI users can have access to LAN-based NetBEUI services. The PPP link becomes the ramp for the remote node to access NetBIOS services on the LAN. (See Figure 23.) An Logical Link Control, type 2 (LLC2) connection is set up between the remote access client and router, and a second LLC2 connection is set up between the router and the remote access (NetBEUI) server.

Figure 23 NetBEUI Connection



By supporting NetBEUI remote clients over PPP, Cisco routers function as a native NetBEUI dial-in router for remote NetBEUI clients. Thus, you can offer remote access to a NetBEUI network through asynchronous or ISDN connections.

To enable a remote access client using a NetBEUI application to connect with the remote router providing NetBEUI services, configure interfaces on the remote access client side and the remote router side by using the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# netbios nbf`	Enables NBFCP on each side of a NetBEUI connection.

To view NetBEUI connection information, use the following command in EXEC mode:

Command	Purpose
`Router> show nbf sessions`	Views NetBEUI connection information.

Configuring Performance Parameters

To tune IP performance, complete the tasks in the following sections:

- Compressing TCP Packet Headers (As required)
- Setting the TCP Connection Attempt Time (As required)
- Compressing IPX Packet Headers over PPP (As required)
- Enabling Fast Switching (As required)
- Controlling Route Cache Invalidation (As required)
- Customizing SLIP and PPP Banner Messages (As required)

Compressing TCP Packet Headers

You can compress the headers of your TCP/IP packets to reduce their size and thereby increase performance. Header compression is particularly useful on networks with a large percentage of small packets, such as those supporting many Telnet connections. This feature compresses only the TCP header, so it has no effect on UDP packets or other protocol headers. The TCP header compression technique, described fully in RFC 1144, is supported on serial lines using High-Level Data Link Control (HDLC) or PPP encapsulation. You must enable compression on both ends of a serial connection.

You can optionally specify outgoing packets to be compressed only when TCP incoming packets on the same interface are compressed. If you do not specify this option, the Cisco IOS software will compress all traffic. The default is no compression.

You can also specify the total number of header compression connections that can exist on an interface. You should configure one connection for each TCP connection through the specified interface.

To enable compression, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip tcp header-compression [passive]`	Enables TCP header compression.
Step 2	`Router(config-if)# ip tcp compression-connections` *number*	Specifies the total number of header compression connections that can exist on an interface.

Note When compression is enabled, fast switching is disabled. Fast processors can handle several fast interfaces, such as T1 lines, that are running header compression. However, you should think carefully about traffic characteristics in your network before compressing TCP headers. You might want to use the monitoring commands to help compare network utilization before and after enabling header compression.

Setting the TCP Connection Attempt Time

You can set the amount of time that the Cisco IOS software will wait to attempt to establish a TCP connection. In previous versions of the Cisco IOS software, the system would wait a fixed 30 seconds when attempting to make the connection. This amount of time is not enough in networks that have dialup asynchronous connections, such as a network consisting of dial-on-demand links that are implemented over modems, because it will affect your ability to use Telnet over the link (from the router) if the link must be brought up.

Because the connection attempt time is a host parameter, it does not pertain to traffic going through the router, just to traffic originated at it.

To set the TCP connection attempt time, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip tcp synwait-time seconds`	Sets the amount of time for which the Cisco IOS software will wait to attempt to establish a TCP connection.

Compressing IPX Packet Headers over PPP

The Cisco IOS software permits compression of IPX packet headers over various WAN media. There are two protocols for IPX compression on point-to-point links:

- CIPX, also known as Telebit style compression
- Shiva compression, which is proprietary

Cisco routers support IPX Header Compression (CIPX) on all point-to-point Novell interfaces over various WAN media.

CIPX is described in RFC 1553, *Compressing IPX Headers Over WAN Media*. The CIPX algorithm is based on the same concepts as Van Jacobson TCP/IP header compression algorithm. CIPX operates over PPP WAN links using either the IPXCP or IPXWAN communications protocols.

CIPX compresses all IPX headers and IPX/NCP headers for Novell packets with the following Network Control Program (NCP) packet types:

- 0x2222—NCP request from workstation
- 0x3333—NCP replies from file server

In this version of software, CIPX is configurable only for PPP links.

CIPX header compression can reduce header information from 30 bytes down to as little as 1 byte. This reduction can save bandwidth and reduce costs associated with IPX routing over WAN links that are configured to use IPXCP or IPXWAN.

Consider the following issues before implementing CIPX:

- CIPX is supported on all point-to-point IPX interfaces using PPP or IPXWAN processing (or both).
- CIPX needs to be negotiated for both directions of the link, because it uses the reverse direction of the link for communicating decompression problems back to the originating peer. In other words, all peer routers must have CIPX enabled.

To configure CIPX, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ipx compression cipx number-of-slots`	Compresses IPX packet headers in a PPP session.

Note We recommend that you keep a slot value of 16. Because slots are maintained in the router buffer, a larger number can impact buffer space for other operations.

Enabling Fast Switching

Fast switching involves the use of a high-speed switching cache for IP routing. With fast switching, destination IP addresses are stored in the high-speed cache so that some time-consuming table lookups can be avoided. The Cisco IOS software generally offers better packet transfer performance when fast switching is enabled.

To enable or disable fast switching, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ip route-cache`	Enables fast-switching (use of a high-speed route cache for IP routing).
Step 2	`Router(config-if)# no ip route-cache`	Disables fast switching and enables load balancing on a per-packet basis.

Controlling Route Cache Invalidation

The high-speed route cache used by IP fast switching is invalidated when the IP routing table changes. By default, the invalidation of the cache is delayed slightly to avoid excessive CPU load while the routing table is changing.

To control route cache invalidation, use the following commands in global configuration mode as needed for your network:

Note This task normally should not be necessary. It should be performed only under the guidance of technical staff. Incorrect configuration can seriously degrade the performance of your router.

	Command	Purpose
Step 1	`Router(config)# no ip cache-invalidate-delay`	Allows immediate invalidation of the cache.
Step 2	`Router(config)# ip cache-invalidate-delay` [*minimum maximum quiet-threshold*]	Delays invalidation of the cache.

Customizing SLIP and PPP Banner Messages

This feature enables you to customize the banner that is displayed when making a SLIP or PPP connection to avoid connectivity problems the default banner message causes in some non-Cisco SLIP and PPP dialup software. This feature is particularly useful when legacy client applications require a specialized connection string.

To configure the SLIP-PPP banner message, use the following command in global configuration mode:

Command	Purpose
`Router(config)# banner slip-ppp` *d message d*	Configures the SLIP-PPP banner to display a customized message.

You can also use tokens in the banner message to display current IOS configuration variables. Tokens are keywords of the form **$(***token***)**. When you include tokens in a banner command, Cisco IOS will replace **$(***token***)** with the corresponding configuration variable.

Table 2 lists the tokens that you can use in the **banner slip-ppp** command.

Table 2 SLIP Banner Tokens

Tokens	Information Displayed in Banner
Global	
$(hostname)	Hostname of the router
$(domain)	Domain name of the router
Slip/PPP Banner-Specific	
$(peer-ip)	IP address of the peer machine
$(gate-ip)	IP address of the gateway machine
$(encap)	Encapsulation type (SLIP, PPP, and so on)
$(encap-alt)	Encapsulation type displayed as SL/IP instead of SLIP
$(mtu)	MTU size

Configuration Examples for Asynchronous SLIP and PPP

This section provides the following examples:

- Basic PPP Configurations Examples
- Remote Node NetBEUI Examples
- Remote Network Access Using PPP Basic Configuration Example
- Remote Network Access Using PPP and Routing IP Example
- Remote Network Access Using a Leased Line with Dial-Backup and PPP Example
- Multilink PPP Using Multiple Asynchronous Interfaces Example

Basic PPP Configurations Examples

The following example illustrates how to make a connection when the system administrator defines a default IP address by including the **peer default ip address** command in interface configuration mode.

Note The **peer default ip address** command replaces the **async default ip address** command.

Once a correct password is entered, you are placed in SLIP mode, and the IP address appears:

```
Router> slip
Password:
Entering SLIP mode.
Your IP address is 192.168.7.28, MTU is 1524 bytes
```

The following example shows the prompts displayed and the response required when dynamic addressing is used to assign the SLIP address:

```
Router> slip
IP address or hostname? 192.168.6.15
Password:
Entering SLIP mode
Your IP address is 192.168.6.15, MTU is 1524 bytes
```

In the previous example, the address 192.168.6.15 had been assigned as the default. Password verification is still required before SLIP mode can be enabled, as follows:

```
Router> slip default
Password:
Entering SLIP mode
Your IP address is 192.168.6.15, MTU is 1524 bytes
```

The following example illustrates the implementation of header compression on the interface with the IP address 172.16.2.1:

```
Router> slip 172.16.2.1 /compressed
Password:
Entering SLIP mode.
Interface IP address is 172.16.2.1, MTU is 1500 bytes.
Header compression will match your system.
```

In the preceding example, the interface is configured for **ip tcp header-compression passive**, which permitted the user to enter the **/compressed** keyword at the EXEC mode prompt. The message "Header compression will match your system" indicates that the user has specified compression. If the line was configured for **ip tcp header-compression on**, this line would read "Header compression is On."

The following example specifies a TACACS server named parlance for address authentication:

```
Router> slip 10.0.0.1@parlance
Password:
Entering SLIP mode.
Interface IP address is 10.0.0.1, MTU is 1500 bytes
Header compression will match your system.
```

The following example sets the SLIP-PPP banner using several tokens and the percent sign (%) as the delimiting character:

```
Router(config)# banner slip-ppp %
Enter TEXT message.  End with the character '%'.
Starting $(encap) connection from $(gate-ip) to $(peer-ip) using a maximum packet size of
$(mtu) bytes... %
```

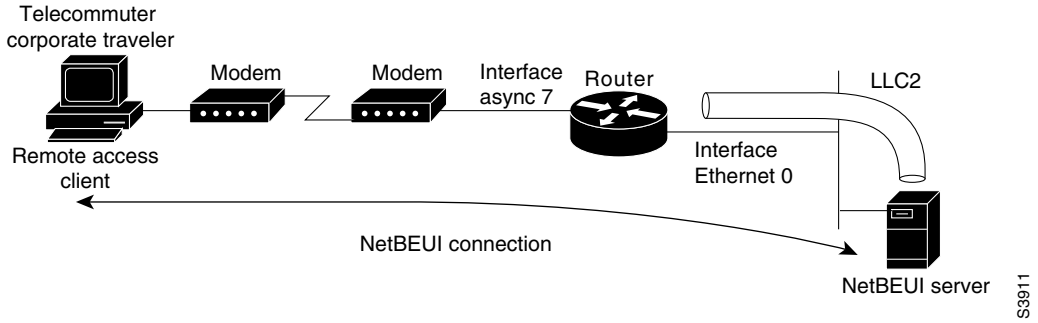
When you enter the **slip** command, you will see the following banner. Notice that the **$(***token***)** syntax is replaced by the corresponding configuration variables.

```
Starting SLIP connection from 192.168.69.96 to 172.16.80.8 using a maximum packet size of
1500 bytes...
```

Remote Node NetBEUI Examples

In the following example, asynchronous interface 7 and Ethernet interface 0 are configured to enable NetBEUI connectivity between the corporate telecommuter client and the remote access (NetBEUI) server. The PC client is running the Chat legacy application in Windows NT to connect with the remote server. (See Figure 24.)

Figure 24 Connecting a Remote NetBEUI Client to a Server Through a Router



The configuration for the router is as follows:

```
interface async 7
 netbios nbf
 encapsulation ppp
```

You would also need to configure security, such as TACACS+, RADIUS, or another form of login authentication on the router.

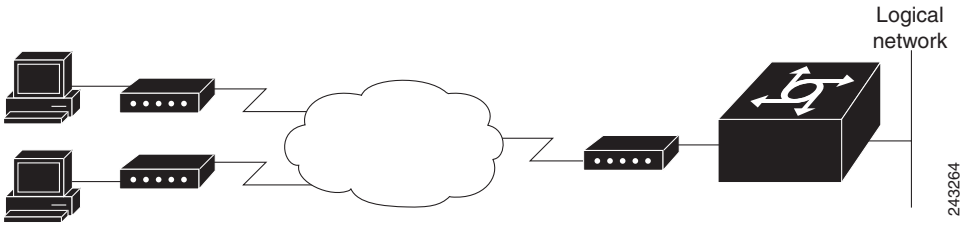
Remote Network Access Using PPP Basic Configuration Example

Figure 25 illustrates a simple network configuration that includes remote PCs with modems connected via modem to a router. The cloud is a Public Switched Telephone Network (PSTN). The modems are connected via asynchronous lines, and the access server is connected to a local network.

In this example, the following is configured:

- An asynchronous line on the access server configured to use PPP encapsulation.
- An interface on the access server for the modem connection; this interface also needs to be configured to accept incoming modem calls.
- A default IP address for each incoming line.

Figure 25 Remote Network Access Using PPP



This default address indicates the address of the remote PC to the server, unless the user explicitly specifies another when starting the PPP session.

The server is configured for interactive mode with autoselect enabled, which allows the user to automatically begin a PPP session upon detection of a PPP packet from the remote PC; or, the remote PC can explicitly begin a PPP session by entering the **ppp** EXEC command at the prompt.

The configuration is as follows:

```
ip routing
```

```
!
interface ethernet 0
 ip address 192.168.32.12 255.255.255.0
!
interface async 1
 encapsulation ppp
 async mode interactive
 async default ip address 192.168.32.51
 async dynamic address
 ip unnumbered ethernet 0

line 1
 autoselect ppp
 modem callin
 speed 19200
```

Remote Network Access Using PPP and Routing IP Example

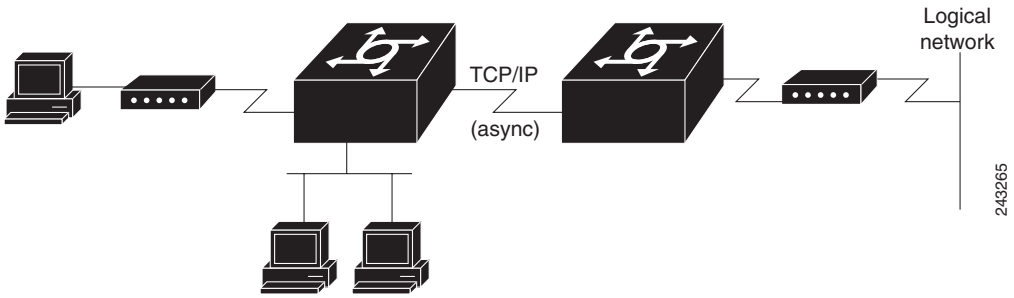
Figure 26 illustrates a network configuration that provides routing functionality, allowing routing updates to be passed across the asynchronous lines.

This network is composed of remote and local PCs connected via modem and network connections to an access server. This access server is connected to a second access server via an asynchronous line running TCP/IP. The second access server is connected to a local network via modem.

For this scenario, you will need to configure the following:

- An asynchronous line on both access servers configured to use PPP encapsulation
- An interface on both access servers for the modem connection and for this interface to be configured to accept incoming modem calls
- A default IP address for each incoming line
- IP routing on all configured interfaces

Figure 26 Routing on an Asynchronous Line Using PPP



The configuration is as follows:

```
interface async 1
 encapsulation ppp
 async mode interactive
 async default ip address 192.168.32.10
 async dynamic address
 ip unnumbered ethernet 0
 async dynamic routing
```

If you want to pass IP routing updates across the asynchronous link, enter the following commands:

```
line 1
 autoselect ppp
 modem callin
 speed 19200
```

Next, enter the following commands to configure the asynchronous lines between the access servers beginning in global configuration mode:

```
interface async 2
 async default ip address 192.168.32.55
 ip tcp header compression passive
```

Finally, configure routing as described in the *Cisco IOS IP Configuration Guide* using one of the following methods. The server can route packets three different ways.

- Use ARP, which is the default behavior.
- Use a default-gateway by entering the command **ip default-gateway** *x.x.x.x,* where *x.x.x.x* is the IP address of a locally attached router.
- Run an IP routing protocol such as Routing Information Protocol (RIP), Interior Gateway Routing Protocol (IGRP), Enhanced IGRP (EIGRP), or Open Shortest Path First (OSPF).

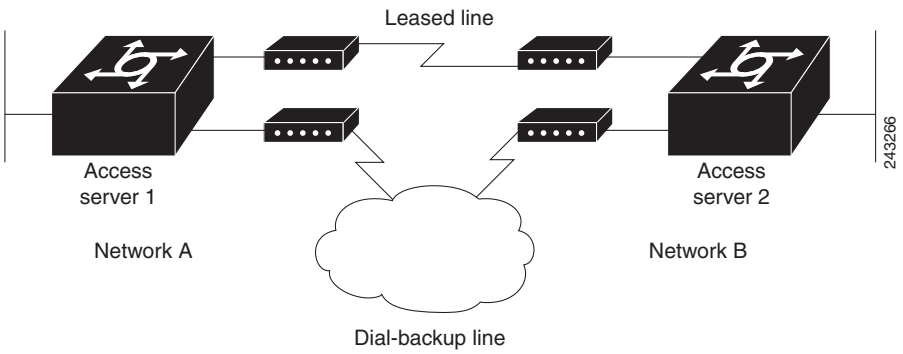
Remote Network Access Using a Leased Line with Dial-Backup and PPP Example

Figure 27 illustrates a scenario where two networks are connected via access servers on a leased line. Redundancy is provided by a dial-backup line over the PSTN so that if the primary leased line goes down, the dial-backup line will be automatically brought up to restore the connection. This configuration would be useful for using an auxiliary port as the backup port for a synchronous port.

For this scenario, you would need to configure the following:

- Two asynchronous interfaces on each access server
- Two modem interfaces
- A default IP address for each interface
- Dial-backup on one modem interface per access server
- An interface connecting to the related network of an access server

Figure 27 Asynchronous Leased Line with Backup



The configuration for this scenario follows:

```
hostname routerA
!
username routerB password cisco
chat-script backup "" "AT" TIMEOUT 30 OK atdt\T TIMEOUT 30 CONNECT \c !
!
interface Serial0
 backup interface Async1
 ip address 192.168.222.12 255.255.255.0
!
interface Async1
 ip address 172.16.199.1 255.255.255.0
 encapsulation ppp
 async default ip address 172.16.199.2
 async dynamic address
 async dynamic routing
 async mode dedicated
 dialer in-band
 dialer map IP 172.16.199.2 name routerB modem-script backup broadcast 3241129
 dialer-group 1
 ppp authentication chap
!
 dialer-list 1 protocol ip permit
!
line aux 0
 modem InOut
 rxspeed 38400
 txspeed 38400
```

Multilink PPP Using Multiple Asynchronous Interfaces Example

The following example shows how to configure MLP using multiple asynchronous interfaces:

```
chat-script backup "" "AT" TIMEOUT 30 OK atdt\T TIMEOUT 30 CONNECT \c
!
ip address-pool local
ip pool foo 10.0.1.5 10.0.1.15
!
int as 1 (2, 3)
 no ip address
 dialer in-band
 encapsulation ppp
 ppp multilink
 dialer-rotary 1
!
interface dialer 1
 encaps ppp
 ip unnumbered ethernet 0
 peer default ip addr pool foo
 ppp authentication chap
 ppp multilink
 dialer in-band
 dialer map ip 10.200.100.9 name WAN-R3 modem-script backup broadcast 2322036
 dialer load-threshold 5 either
 dialer-group 1
!
dialer-list 1 protocol ip permit
!
line line 1 3
modem InOut
speed 115000
```

Configuring Media-Independent PPP and Multilink PPP

Configuring Media-Independent PPP and Multilink PPP

This chapter describes how to configure the PPP and Multilink PPP (MLP) features that can be configured on any interface. It includes the following main sections:

This chapter also describes address pooling for point-to-point links, which is available on all asynchronous serial, synchronous serial, and ISDN interfaces. See the chapter "Configuring Asynchronous SLIP and PPP" in this publication for information about PPP features and requirements that apply only to asynchronous lines and interfaces.

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

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

PPP Encapsulation Overview

PPP, described in RFC 1661, encapsulates network layer protocol information over point-to-point links. You can configure PPP on the following types of physical interfaces:

- Asynchronous serial
- High-Speed Serial Interface (HSSI)
- ISDN

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

- Synchronous serial

Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP)Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the **down-when-looped** command is configured, the router might shut down a link if it detects a loop.

The software provides the CHAP and PAP on serial interfaces running PPP encapsulation. For detailed information about authentication, refer to the *Cisco IOS Security Configuration Guide*.

Beginning with Cisco IOS Release 11.2 F, Cisco supported fast switching of incoming and outgoing DECnet and CLNS packets over PPP.

Configuring PPP and MLP

To configure PPP on a serial interface (including ISDN), perform the following task in interface configuration mode. This task is required for PPP encapsulation.

- Enabling PPP Encapsulation

You can also complete the tasks in the following sections; these tasks are optional but offer a variety of uses and enhancements for PPP on your systems and networks:

- Enabling CHAP or PAP Authentication
- Enabling Link Quality Monitoring
- Configuring Compression of PPP Data
- Configuring Microsoft Point-to-Point Compression
- Configuring IP Address Pooling
- Configuring PPP Reliable Link
- Disabling or Reenabling Peer Neighbor Routes
- Configuring PPP Half-Bridging
- Configuring Multilink PPP
- Configuring MLP Interleaving
- Enabling Distributed CEF Switching
- Creating a Multilink Bundle
- Assigning an Interface to a Multilink Bundle
- Disabling PPP Multilink Fragmentation
- Verifying the MLP Inverse Multiplexer Configuration

See the section "Monitoring and Maintaining PPP and MLP Interfaces" later in this chapter for tips on maintaining PPP. See the "Configuration Examples for PPP and MLP" at the end of this chapter for ideas on how to implement PPP and MLP in your network.

Enabling PPP Encapsulation

To enable PPP on serial lines to encapsulate IP and other network protocol datagrams, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.

Enabling CHAP or PAP Authentication

PPP with CHAP or PAP authentication is often used to inform the central site about which remote routers are connected to it.

With this authentication information, if the router or access server receives another packet for a destination to which it is already connected, it does not place an additional call. However, if the router or access server is using rotaries, it sends the packet out the correct port.

CHAP and PAP were originally specified in RFC 1334, and CHAP is updated in RFC 1994. These protocols are supported on synchronous and asynchronous serial interfaces. When using CHAP or PAP authentication, each router or access server identifies itself by a *name*. This identification process prevents a router from placing another call to a router to which it is already connected, and also prevents unauthorized access.

Access control using CHAP or PAP is available on all serial interfaces that use PPP encapsulation. The authentication feature reduces the risk of security violations on your router or access server. You can configure either CHAP or PAP for the interface.

Note To use CHAP or PAP, you must be running PPP encapsulation.

When CHAP is enabled on an interface and a remote device attempts to connect to it, the local router or access server sends a CHAP packet to the remote device. The CHAP packet requests or "challenges" the remote device to respond. The challenge packet consists of an ID, a random number, and the host name of the local router.

The required response has two parts:

- An encrypted version of the ID, a secret password, and the random number
- Either the host name of the remote device or the name of the user on the remote device

When the local router or access server receives the response, it verifies the secret password by performing the same encryption operation as indicated in the response and looking up the required host name or username. The secret passwords must be identical on the remote device and the local router.

Because this response is sent, the password is never sent in clear text, preventing other devices from stealing it and gaining illegal access to the system. Without the proper response, the remote device cannot connect to the local router.

CHAP transactions occur only when a link is established. The local router or access server does not request a password during the rest of the call. (The local device can, however, respond to such requests from other devices during a call.)

When PAP is enabled, the remote router attempting to connect to the local router or access server is required to send an authentication request. If the username and password specified in the authentication request are accepted, the Cisco IOS software sends an authentication acknowledgment.

After you have enabled CHAP or PAP, the local router or access server requires authentication from remote devices. If the remote device does not support the enabled protocol, no traffic will be passed to that device.

To use CHAP or PAP, you must perform the following tasks:

- Enable PPP encapsulation.
- Enable CHAP or PAP on the interface.
- For CHAP, configure host name authentication and the secret or password for each remote system with which authentication is required.

To enable PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation on an interface.

To enable CHAP or PAP authentication on an interface configured for PPP encapsulation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp authentication {chap | chap pap | pap chap | pap} [if-needed] [list-name | default] [callin]`	Defines the authentication methods supported and the order in which they are used.

The **ppp authentication chap** optional keyword **if-needed** can be used only with Terminal Access Controller Access Control System (TACACS) or extended TACACS.

With authentication, authorization, and accounting (AAA) configured on the router and list names defined for AAA, the *list-name* optional keyword can be used with AAA/TACACS+.

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

Add a **username** entry for each remote system from which the local router or access server requires authentication.

To specify the password to be used in CHAP or PAP caller identification, use the following command in global configuration mode:

Command	Purpose
`Router(config)# username name [user-maxlinks link-number] password secret`	Configures identification. Optionally, you can specify the maximum number of connections a user can establish. To use the **user-maxlinks** keyword, you must also use the **aaa authorization network default local** command and PPP encapsulation and name authentication on all the interfaces the user will be accessing.

Make sure this password does not include spaces or underscores.

To configure TACACS on a specific interface as an alternative to global host authentication, use one of the following commands in interface configuration mode:

Command	Purpose

`Router(config-if)# ppp use-tacacs [single-line]`
or
`Router(config-if)# aaa authentication ppp`
	Configures TACACS.

Use the **ppp use-tacacs** command with TACACS and Extended TACACS. Use the **aaa authentication ppp** command with AAA/TACACS+.

For an example of CHAP, see the section "CHAP with an Encrypted Password Examples" at the end of this chapter. CHAP is specified in RFC 1994, *PPP Challenge Handshake Authentication Protocol (CHAP)*.

Enabling Link Quality Monitoring

Link Quality Monitoring (LQM) is available on all serial interfaces running PPP. LQM will monitor the link quality, and if the quality drops below a configured percentage, the router will shut down the link. The percentages are calculated for both the incoming and outgoing directions. The outgoing quality is calculated by comparing the total number of packets and bytes sent with the total number of packets and bytes received by the destination node. The incoming quality is calculated by comparing the total number of packets and bytes received with the total number of packets and bytes sent by the destination peer.

Note LQM is not compatible with Multilink PPP.

When LQM is enabled, Link Quality Reports (LQRs) are sent, in place of keepalives, every keepalive period. All incoming keepalives are responded to properly. If LQM is not configured, keepalives are sent every keepalive period and all incoming LQRs are responded to with an LQR.

LQR is specified in RFC 1989, *PPP Link Quality Monitoring*.

To enable LQM on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp quality` *percentage*	Enables LQM on the interface.

The *percentage* argument specifies the link quality threshold. That percentage must be maintained, or the link is deemed to be of poor quality and is taken down.

Configuring Compression of PPP Data

You can configure point-to-point software compression on serial interfaces that use PPP encapsulation. Compression reduces the size of a PPP frame via lossless data compression. PPP encapsulations support both predictor and Stacker compression algorithms.

If most of your traffic is already compressed files, do not use compression.

Most routers support software compression only, but in the Cisco 7000 series routers, hardware compression and distributed compression are also available, depending on the interface processor and compression service adapter hardware installed in the router.

To configure compression, complete the tasks in one of the following sections:

- Software Compression
- Hardware-Dependent Compression

Software Compression

Software compression is available in all router platforms. Software compression is performed by the main processor in the router.

Compression is performed in software and might significantly affect system performance. We recommend that you disable compression if the router CPU load exceeds 65 percent. To display the CPU load, use the **show process cpu** EXEC command.

To configure compression over PPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.
Step 2	`Router(config-if)# compress [predictor | stac | mppc [ignore-pfc]]`	Enables compression.

Hardware-Dependent Compression

When you configure Stacker compression on Cisco 7000 series routers with a 7000 Series Route Switch Processor (RSP7000), on Cisco 7200 series routers, and on Cisco 7500 series routers, there are three methods of compression: hardware compression, distributed compression, and software compression.

Hardware and distributed compression are available on routers that have the SA-Comp/1 and SA-Comp/4 data compression service adapters (CSAs). CSAs are available on Cisco 7200 series routers, on Cisco 7500 series routers with second-generation Versatile Interface Processors (VIP2s), and on Cisco 7000 series routers with the RSP7000 and 7000 Series Chassis Interface (RSP7000CI). (CSAs require VIP2 model VIP2-40.)

To configure hardware or distributed compression over PPP, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# encapsulation ppp`	Enables encapsulation of a single protocol on the serial line.

Step 2	**Cisco 7000 series with RSP7000 and Cisco 7500 series routers**
	`Router(config-if)# compress stac [distributed | software]`
	Cisco 7200 series routers
	`Router(config-if)# compress stac [csa slot | software]`
		Enables compression.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method:

- If the router contains a CSA, compression is performed in the CSA hardware (hardware compression).

- If the CSA is not available, compression is performed in the software installed on the VIP2 (distributed compression).
- If the VIP2 is not available, compression is performed in the main processor of the router (software compression).

Using hardware compression in the CSA frees the main processor of the router for other tasks. You can also configure the router to use the VIP2 to perform compression by using the **distributed** option, or to use the main processor of the router by using the **software** option. If the VIP2 is not available, compression is performed in the main processor of the router.

When compression is performed in software installed in the main processor of the router, it might substantially affect system performance. We recommend that you disable compression in the main processor of the router if the router CPU load exceeds 40 percent. To display the CPU load, use the **show process cpu** EXEC command.

Specifying the **compress stac** command with no options causes the router to use the fastest available compression method.

Configuring Microsoft Point-to-Point Compression

Microsoft Point-to-Point Compression (MPPC) is a scheme used to compress PPP packets between Cisco and Microsoft client devices. The MPPC algorithm is designed to optimize bandwidth utilization in order to support multiple simultaneous connections. The MPPC algorithm uses a Lempel-Ziv (LZ)-based algorithm with a continuous history buffer called a dictionary.

The Compression Control Protocol (CCP) configuration option for MPPC is 18.

Exactly one MPPC datagram is encapsulated in the PPP information field. The PPP protocol field indicates the hexadecimal type of 00FD for all compressed datagrams. The maximum length of the MPPC datagram sent over PPP is the same as the MTU of the PPP interface; however, this length cannot be greater than 8192 bytes because the history buffer is limited to 8192 bytes. If compressing the data results in data expansion, the original data is sent as an uncompressed MPPC packet.

The history buffers between compressor and decompressor are synchronized by maintaining a 12-bit coherency count. If the decompressor detects that the coherency count is out of sequence, the following error recovery process is performed:

1. Reset Request (RR) packet is sent from the decompressor.
2. The compressor then flushes the history buffer and sets the flushed bit in the next packet it sends.
3. Upon receiving the flushed bit set packet, the decompressor flushes the history buffer.

Synchronization is achieved without CCP using the Reset Acknowledge (RA) packet, which can consume additional time.

Compression negotiation between a router and a Windows 95 client occurs through the following process:

1. Windows 95 sends a request for both STAC (option 17) and MPPC (option 18) compression.
2. The router sends a negative acknowledgment (NAK) requesting only MPPC.
3. Windows 95 resends the request for MPPC.
4. The router sends an acknowledgment (ACK) confirming MPPC compression negotiation.

MPPC Restrictions

The following restrictions apply to the MPPC feature:

- MPPC is supported only with PPP encapsulation.
- Compression can be processor intensive because it requires a reserved block of memory to maintain the history buffer. Do not enable modem or hardware compression because it may cause performance degradation, compression failure, or data expansion.
- Both ends of the point-to-point link must be using the same compression method (STAC, Predictor, or MPPC, for example).

Configuring MPPC

PPP encapsulation must be enabled before you can configure MPPC. For information on how to configure PPP encapsulation, see the section "Enabling PPP Encapsulation" earlier in this chapter.

There is only one command required to configure MPPC. The existing **compress** command supports the **mppc** keyword, which prepares the interface to initiate CCP and negotiates MPPC with the Microsoft client. To set MPPC once PPP encapsulation is configured on the router, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# compress [mppc [ignore-pfc]]`	Enables MPPC on the interface.

The **ignore-pfc** keyword instructs the router to ignore the protocol field compression flag negotiated by LCP. For example, the uncompressed standard protocol field value for IP is 0x0021 and 0x21 when compression is enabled. When the **ignore-pfc** option is enabled, the router will continue to use the uncompressed value (0x0021). Using the **ignore-pfc** option is helpful for some asynchronous driver devices that use an uncompressed protocol field (0x0021), even though the protocol field compression is negotiated between peers. displays protocol rejections when the **debug ppp negotiation** command is enabled. These errors can be remedied by setting the **ignore-pfc** option.

Sample debug ppp negotiation Command Output Showing Protocol Reject

```
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
PPP Async2: protocol reject received for protocol = 0x2145
```

Configuring IP Address Pooling

A point-to-point interface must be able to provide a remote node with its IP address through the IP Control Protocol (IPCP) address negotiation process. The IP address can be obtained from a variety of sources. The address can be configured through the command line, entered with an EXEC-level command, provided by TACACS+ or the Dynamic Host Configuration Protocol (DHCP), or from a locally administered pool.

IP address pooling uses a pool of IP addresses from which an incoming interface can provide an IP address to a remote node through IPCP address negotiation process. IP address pooling also enhances configuration flexibility by allowing multiple types of pooling to be active simultaneously.

See the chapter "Configuring Asynchronous SLIP and PPP" in this publication for additional information about address pooling on asynchronous interfaces and about the Serial Line Internet Protocol (SLIP).

Peer Address Allocation

A peer IP address can be allocated to an interface through several methods:

- Dialer map lookup—This method is used only if the peer requests an IP address, no other peer IP address has been assigned, and the interface is a member of a dialer group.
- PPP or SLIP EXEC command—An asynchronous dialup user can enter a peer IP address or host name when PPP or SLIP is invoked from the command line. The address is used for the current session and then discarded.
- IPCP negotiation—If the peer presents a peer IP address during IPCP address negotiation and no other peer address is assigned, the presented address is acknowledged and used in the current session.
- Default IP address—The **peer default ip address** command and the **member peer default ip address** command can be used to define default peer IP addresses.
- TACACS+ assigned IP address—During the authorization phase of IPCP address negotiation, TACACS+ can return an IP address that the user being authenticated on a dialup interface can use. This address overrides any default IP address and prevents pooling from taking place.
- DHCP retrieved IP address—If configured, the routers acts as a proxy client for the dialup user and retrieves an IP address from a DHCP server. That address is returned to the DHCP server when the timer expires or when the interface goes down.
- Local address pool—The local address pool contains a set of contiguous IP addresses (a maximum of 1024 addresses) stored in two queues. The free queue contains addresses available to be assigned and the used queue contains addresses that are in use. Addresses are stored to the free queue in first-in, first-out (FIFO) order to minimize the chance the address will be reused, and to allow a peer to reconnect using the same address that it used in the last connection. If the address is available, it is assigned; if not, another address from the free queue is assigned.
- Chat script (asynchronous serial interfaces only)—The IP address in the **dialer map** command entry that started the script is assigned to the interface and overrides any previously assigned peer IP address.
- Virtual terminal/protocol translation—The translate command can define the peer IP address for a virtual terminal (pseudo asynchronous interface).
- The pool configured for the interface is used, unless TACACS+ returns a pool name as part of AAA. If no pool is associated with a given interface, the global pool named default is used.

Precedence Rules

The following precedence rules of peer IP address support determine which address is used. Precedence is listed from most likely to least likely:

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

5. Configured address from the **peer default ip address** command or address from the protocol **translate** command
6. Peer provided address from IPCP negotiation (not accepted unless no other address exists)

Interfaces Affected

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

Choosing the IP Address Assignment Method

The IP address pooling feature now allows configuration of a global default address pooling mechanism, per-interface configuration of the address pooling mechanism, and per-interface configuration of a specific address or pool name.

You can define the type of IP address pooling mechanism used on router interfaces in one or both of the ways described in the following sections:

- Defining the Global Default Address Pooling Mechanism
- Configuring IP Address Assignment

Defining the Global Default Address Pooling Mechanism

The global default mechanism applies to all point-to-point interfaces that support PPP encapsulation and that have not otherwise been configured for IP address pooling. You can define the global default mechanism to be either DHCP or local address pooling.

To configure the global default mechanism for IP address pooling, perform the tasks in one of following sections:

- Defining DHCP as the Global Default Mechanism
- Defining Local Address Pooling as the Global Default Mechanism

After you have defined a global default mechanism, you can disable it on a specific interface by configuring the interface for some other pooling mechanism. You can define a local pool other than the default pool for the interface or you can configure the interface with a specific IP address to be used for dial-in peers.

You can also control the DHCP network discovery mechanism; see the following section for more information:

- Controlling DHCP Network Discovery

Defining DHCP as the Global Default Mechanism

DHCP specifies the following components:

- A DHCP server—A host-based DHCP server configured to accept and process requests for temporary IP addresses.
- A DHCP proxy-client—A Cisco access server configured to arbitrate DHCP calls between the DHCP server and the DHCP client. The DHCP client-proxy feature manages a pool of IP addresses available to dial-in clients without a known IP address.

To enable DHCP as the global default mechanism, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip address-pool dhcp-proxy-client`	Specifies DHCP client-proxy as the global default mechanism.
Step 2	`Router(config)# ip dhcp-server [ip-address | name]`	(Optional) Specifies the IP address of a DHCP server for the proxy client to use.

In Step 2, you can provide as few as one or as many as ten DHCP servers for the proxy-client (the Cisco router or access server) to use. DHCP servers provide temporary IP addresses.

Defining Local Address Pooling as the Global Default Mechanism

To specify that the global default mechanism to use is local pooling, use the following commands in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip address-pool local`	Specifies local pooling as the global default mechanism.
Step 2	`Router(config)# ip local pool {named-address-pool | default} {first-IP-address [last-IP-address]} [group group-name] [cache-size size]}`	Creates one or more local IP address pools.

If no other pool is defined, a local pool called “default” is used. Optionally, you can associate an address pool with a named pool group.

Controlling DHCP Network Discovery

To allow peer routers to dynamically discover Domain Name System (DNS) and NetBIOS name server information configured on a DHCP server using PPP IP Control Protocol (IPCP) extensions, use the following command in global configuration mode:

Command	Purpose
`Router(config)# ip dhcp-client network-discovery informs number-of-messages discovers number-of-messages period seconds`	Provides control of the DHCP network discovery mechanism by allowing the number of DHCP Inform and Discover messages to be sent, and a time-out period for retransmission, to be configured.

The **ip dhcp-client network-discovery** global configuration command provides a way to control the DHCP network discovery mechanism. The number of DHCP Inform or Discovery messages can be set to 1 or 2, which determines how many times the system sends the DHCP Inform or Discover messages before stopping network discovery. You can set a time-out period from 3 to 15 seconds, or leave the default time-out period at 15 seconds. Default for the **informs** and **discovers** keywords is 0, which disables the transmission of these messages.

Configuring IP Address Assignment

After you have defined a global default mechanism for assigning IP addresses to dial-in peers, you can configure the few interfaces for which it is important to have a nondefault configuration. You can do any of the following;

- Define a nondefault address pool for use by a specific interface.
- Define DHCP on an interface even if you have defined local pooling as the global default mechanism.
- Specify one IP address to be assigned to all dial-in peers on an interface.
- Make temporary IP addresses available on a per-interface basis to asynchronous clients using SLIP or PPP.

To define a nondefault address pool for use on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# ip local pool {named-address-pool | default} {first-IP-address [last-IP-address]} [group group-name] [cache-size size]`	Creates one or more local IP address pools.
Step 2	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 3	`Router(config-if)# peer default ip address pool pool-name-list`	Specifies the pool or pools for the interface to use.

To define DHCP as the IP address mechanism for an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 2	`Router(config-if)# peer default ip address pool dhcp`	Specifies DHCP as the IP address mechanism on this interface.

To define a specific IP address to be assigned to all dial-in peers on an interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and begins interface configuration mode.
Step 2	`Router(config-if)# peer default ip address ip-address`	Specifies the IP address to assign.

Configuring PPP Reliable Link

PPP reliable link is Cisco's implementation of RFC 1663, *PPP Reliable Transmission*, which defines a method of negotiating and using Numbered Mode Link Access Procedure, Balanced (LAPB) to provide a reliable serial link. Numbered Mode LAPB provides retransmission of error packets across the serial link.

Although LAPB protocol overhead consumes some bandwidth, you can offset that consumption by the use of PPP compression over the reliable link. PPP compression is separately configurable and is not required for use of a reliable link.

Note PPP reliable link is available only on synchronous serial interfaces, including ISDN BRI and ISDN PRI interfaces. PPP reliable link cannot be used over V.120, and does not work with Multilink PPP.

To configure PPP reliable link on a specified interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp reliable-link`	Enables PPP reliable link.

Having reliable links enabled does not guarantee that all connections through the specified interface will in fact use reliable link. It only guarantees that the router will attempt to negotiate reliable link on this interface.

Troubleshooting PPP

You can troubleshoot PPP reliable link by using the **debug lapb** command and the **debug ppp negotiations**, **debug ppp errors**, and **debug ppp packets** commands. You can determine whether LAPB has been established on a connection by using the **show interface** command.

Disabling or Reenabling Peer Neighbor Routes

The Cisco IOS software automatically creates neighbor routes by default; that is, it automatically sets up a route to the peer address on a point-to-point interface when the PPP IPCP negotiation is completed.

To disable this default behavior or to reenable it once it has been disabled, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no peer neighbor-route`	Disables creation of neighbor routes.
Step 2	`Router(config-if)# peer neighbor-route`	Reenables creation of neighbor routes.

Note If entered on a dialer or asynchronous group interface, this command affects all member interfaces.

Configuring PPP Half-Bridging

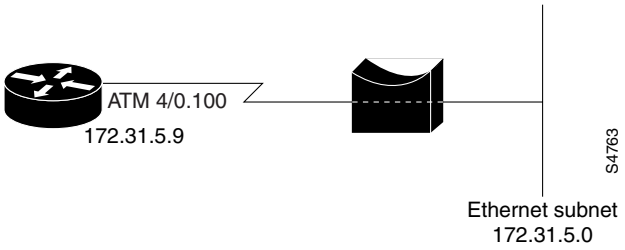
For situations in which a routed network needs connectivity to a remote bridged Ethernet network, a serial or ISDN interface can be configured to function as a PPP half-bridge. The line to the remote bridge functions as a virtual Ethernet interface, and the serial or ISDN interface on the router functions as a node on the same Ethernet subnetwork as the remote network.

The bridge sends bridge packets to the PPP half-bridge, which converts them to routed packets and forwards them to other router processes. Likewise, the PPP half-bridge converts routed packets to Ethernet bridge packets and sends them to the bridge on the same Ethernet subnetwork.

Note An interface cannot function as both a half-bridge and a bridge.

Figure 91 shows a router with a serial interface configured as a PPP half-bridge. The interface functions as a node on the Ethernet subnetwork with the bridge. Note that the serial interface has an IP address on the same Ethernet subnetwork as the bridge.

Figure 91 Router Serial Interface Configured as a Half-Bridge



Note The Cisco IOS software supports no more than one PPP half-bridge per Ethernet subnetwork.

To configure a serial interface to function as a half-bridge, use the following commands beginning in global configuration mode as appropriate for your network:

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies the interface and begins interface configuration mode.

Step 2	`Router(config-if)# ppp bridge appletalk`
	`Router(config-if)# ppp bridge ip`
	`Router(config-if)# ppp bridge ipx [novell-ether | arpa | sap | snap]`
		Enables PPP half-bridging for one or more routed protocols: AppleTalk, IP, or Internet Protocol Exchange (IPX).

Step 3	`Router(config-if)# ip address n.n.n.n`
	`Router(config-if)# appletalk address network.node`
	`Router(config-if)# appletalk cable-range cable-range network.node`
	`Router(config-if)# ipx network network`
		Provides a protocol address on the same subnetwork as the remote network.

Note You must enter the **ppp bridge** command either when the interface is shut down or before you provide a protocol address for the interface.

For more information about AppleTalk addressing, refer to the "Configuring AppleTalk" chapter of the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*. For more information about IPX addresses and encapsulations, refer to the "Configuring Novell IPX" chapter of the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

Configuring Multilink PPP

The Multilink PPP feature provides load balancing functionality over multiple WAN links, while providing multivendor interoperability, packet fragmentation and proper sequencing, and load calculation on both inbound and outbound traffic. The Cisco implementation of MLP supports the fragmentation and packet sequencing specifications in RFC 1990. Additionally, you can change the default endpoint discriminator value that is supplied as part of user authentication. Refer to RFC 1990 for more information about the endpoint discriminator.

MLP allows packets to be fragmented and the fragments to be sent at the same time over multiple point-to-point links to the same remote address. The multiple links come up in response to a defined dialer load threshold. The load can be calculated on inbound traffic, outbound traffic, or on either, as needed for the traffic between the specific sites. MLP provides bandwidth on demand and reduces transmission latency across WAN links.

MLP is designed to work over synchronous and asynchronous serial and BRI and PRI types of single or multiple interfaces that have been configured to support both dial-on-demand rotary groups and PPP encapsulation.

Perform the tasks in the following sections, as required for your network, to configure MLP:

- Configuring MLP on Synchronous Interfaces
- Configuring MLP on Asynchronous Interfaces
- Configuring MLP on a Single ISDN BRI Interface
- Configuring MLP on Multiple ISDN BRI Interfaces
- Configuring MLP Using Multilink Group Interfaces
- Changing the Default Endpoint Discriminator

Configuring MLP on Synchronous Interfaces

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

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

	Command	Purpose
Step 1	`Router(config)# interface serial number`	Specifies an asynchronous interface.
Step 2	`Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# no fair-queue`	Disables WFQ on the interface.
Step 5	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 6	`Router(config-if)# pulse-time seconds`	Enables pulsing DTR signal intervals on the interface.

Repeat these steps for additional synchronous interfaces, as needed.

Configuring MLP on Asynchronous Interfaces

To configure MLP on asynchronous interfaces, configure the asynchronous interfaces to support dial-on-demand routing (DDR) and PPP encapsulation, and then configure a dialer interface to support PPP encapsulation, bandwidth on demand, and Multilink PPP.

To configure an asynchronous interface to support DDR and PPP encapsulation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface async number`	Specifies an asynchronous interface and begins interface configuration mode.
Step 2	`Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer in-band`	Enables DDR on the interface.
Step 5	`Router(config-if)# dialer rotary-group number`	Includes the interface in a specific dialer rotary group.

Repeat these steps for additional asynchronous interfaces, as needed.

At some point, adding more asynchronous interfaces does not improve performance, With the default maximum transmission unit (MTU) size, MLP should support three asynchronous interfaces using V.34 modems. However, packets might be dropped occasionally if the maximum transmission unit (MTU) size is small or large bursts of short frames occur.

To configure a dialer interface to support PPP encapsulation and Multilink PPP, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Defines a dialer rotary group.
Step 2	`Router(config-if)# no ip address`	Specifies no IP address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer in-band`	Enables DDR on the interface.
Step 5	`Router(config-if)# dialer load-threshold load [inbound | outbound | either]`	Configures bandwidth on demand by specifying the maximum load before the dialer places another call to a destination.
Step 6	`Router(config-if)# ppp multilink`	Enables Multilink PPP.

Configuring MLP on a Single ISDN BRI Interface

To enable MLP on a single ISDN BRI interface, you are not required to define a dialer rotary group separately because ISDN interfaces are dialer rotary groups by default.

To enable PPP on an ISDN BRI interface, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri number`	Specifies an interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address ip-address mask [secondary]`	Provides an appropriate protocol address for the interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer idle-timeout seconds [inbound | either]`	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 5	`Router(config-if)# dialer load-threshold load`	Specifies the dialer load threshold for bringing up additional WAN links.
Step 6	`Router(config-if)# dialer map protocol next-hop-address [name hostname] [spc] [speed 56 | 64] [broadcast] [dial-string[:isdn-subaddress]]`	Configures the ISDN interface to call the remote site.
Step 7	`Router(config-if)# dialer-group group-number`	Controls access to this interface by adding it to a dialer access group.
Step 8	`Router(config-if)# ppp authentication pap`	(Optional) Enables PPP authentication.
Step 9	`Router(config-if)# ppp multilink`	Enables MLP on the dialer rotary group.

If you do not use PPP authentication procedures (Step 8), your telephone service must pass caller ID information.

The load threshold number is required. For an example of configuring MLP on a single ISDN BRI interface, see the section "MLP on One ISDN BRI Interface Example" at the end of this chapter.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer-load threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely, and the **dialer-load threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Configuring MLP on Multiple ISDN BRI Interfaces

To enable MLP on multiple ISDN BRI interfaces, set up a dialer rotary interface and configure it for Multilink PPP, and then configure the BRI interfaces separately and add them to the same rotary group.

To set up the dialer rotary interface for the BRI interfaces, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface dialer number`	Specifies the dialer rotary interface and begins interface configuration mode.
Step 2	`Router(config-if)# ip address address mask`	Specifies the protocol address for the dialer rotary interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.

	Command	Purpose
Step 4	`Router(config-if)# dialer in-band`	Specifies in-band dialing.
Step 5	`Router(config-if)# dialer idle-timeout` *seconds* [**inbound** | **either**]	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, both inbound and outbound traffic will reset the dialer idle timer. Including the **inbound** keyword will cause only inbound traffic to reset the timer.
Step 6	`Router(config-if)# dialer map` *protocol next-hop-address* [**name** *hostname*] [**spc**] [**speed 56** | **64**] [**broadcast**] [*dial-string*[**:***isdn-subaddress*]]	Maps the next hop protocol address and name to the dial string needed to reach it.
Step 7	`Router(config-if)# dialer load-threshold` *load*	Specifies the dialer load threshold, using the same threshold as the individual BRI interfaces.
Step 8	`Router(config-if)# dialer-group` *number*	Controls access to this interface by adding it to a dialer access group.
Step 9	`Router(config-if)# ppp authentication chap`	(Optional) Enables PPP CHAP authentication.
Step 10	`Router(config-if)# ppp multilink`	Enables Multilink PPP.

If you do not use PPP authentication procedures (Step 10), your telephone service must pass caller ID information.

To configure each of the BRI interfaces to belong to the same rotary group, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface bri` *number*	Specifies one of the BRI interfaces.
Step 2	`Router(config-if)# no ip address`	Specifies that it does not have an individual protocol address.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# dialer idle-timeout` *seconds* [**inbound** | **either**]	Specifies the duration of idle time in seconds after which a line will be disconnected. By default, outbound traffic will reset the dialer idle timer. Adding the **either** keyword causes both inbound and outbound traffic to reset the timer; adding the **inbound** keyword causes only inbound traffic to reset the timer.
Step 5	`Router(config-if)# dialer rotary-group` *number*	Adds the interface to the rotary group.
Step 6	`Router(config-if)# dialer load-threshold` *load*	Specifies the dialer load threshold for bringing up additional WAN links.

Repeat Steps 1 through 6 for each BRI that you want to belong to the same dialer rotary group.

When MLP is configured and you want a multilink bundle to be connected indefinitely, use the **dialer idle-timeout** command to set a very high idle timer. (The **dialer load-threshold 1** command no longer keeps a multilink bundle of *n* links connected indefinitely and the **dialer load-threshold 2** command no longer keeps a multilink bundle of two links connected indefinitely.)

Note Previously, when MLP was used in a dialer profile, a virtual access interface was always created as the bundle. It was bound to both the B channel and the dialer profile interfaces after creation and cloning. The dialer profile interface could act as the bundle without help from a virtual access interface. But with the Dynamic Multiple Encapsulations feature available in Cisco IOS Release 12.1, it is no longer the virtual access interface that is added into the connected group of the dialer profile, but the dialer profile itself. The dialer profile becomes a connected member of its own connected group. See the "Dynamic Multiple Encapsulations over ISDN Example" in the chapter "Configuring Peer-to-Peer DDR with Dialer Profiles" in this publication, for more information about dynamic multiple encapsulations and its relation to Multilink PPP.

For an example of configuring MLP on multiple ISDN BRI interfaces, see the section "MLP on Multiple ISDN BRI Interfaces Example" at the end of this chapter.

Configuring MLP Using Multilink Group Interfaces

MLP can be configured by assigning a multilink group to a virtual template configuration. Virtual templates allow a virtual access interface to dynamically clone interface parameters from the specified virtual template. If a multilink group is assigned to a virtual template, and then the virtual template is assigned to a physical interface, all links that pass through the physical interface will belong to the same multilink bundle.

A multilink group interface configuration will override a global multilink virtual template configured with the **multilink virtual template** command.

Multilink group interfaces can be used with ATM, PPP over Frame Relay, and serial interfaces.

To configure MLP using a multilink group interface, perform the following tasks:

- Configure the multilink group.
- Assign the multilink group to a virtual template.
- Configure the physical interface to use the virtual template.

To configure the multilink group, use the following commands beginning in global configuration mode:

Command	Purpose
`Router(config)# interface multilink group-number`	Creates a multilink bundle and enters multilink interface configuration mode to configure the bundle.
`Router(config-if)# ip address address mask`	Sets a primary IP address for an interface.
`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
`Router(config-if)# ppp multilink`	Enables MLP on an interface.

To assign the multilink group to a virtual template, perform the following task beginning in global configuration mode:

`Router(config)# interface virtual template number`	Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces.
`Router(config-if)# ppp multilink group group-number`	Restricts a physical link to joining only a designated multilink-group interface.

To configure the physical interface and assign the virtual template to it, perform the following task beginning in global configuration mode. This example is for an ATM interface. However, multilink group interfaces can also be used with PPP over Frame Relay interfaces and serial interfaces.

`Router(config)# interface atm` *interface-number.subinterface-number* `point-to-point`	Configures an ATM interface and enters interface configuration mode.
`Router(config-if)# pvc` *vpi/vci*	Creates or assigns a name to an ATM permanent virtual circuit (PVC), specifies the encapsulation type on an ATM PVC, and enters ATM virtual circuit configuration mode.
`Router(config-if-atm-vc)# protocol ppp virtual-template` *name*	Configures VC multiplexed encapsulation on a PVC.

To see an example of how to configure MLP over an ATM PVC using a multilink group, see the section "MLP Using Multilink Group Interfaces over ATM Example" at the end of this chapter.

Changing the Default Endpoint Discriminator

By default, when the system negotiates use of MLP with the peer, the value that is supplied for the endpoint discriminator is the same as the username used for authentication. That username is configured for the interface by the Cisco IOS **ppp chap hostname** or **ppp pap sent-username** command, or defaults to the globally configured host name (or stack group name, if this interface is a Stack Group Bidding Protocol, or SGBP, group member).

To override or change the default endpoint discriminator, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink endpoint {hostname | ip` *IP-address* `| mac` *LAN-interface* `| none | phone` *telephone-number* `| string` *char-string*`}`	Overrides or changes the default endpoint discriminator the system uses when negotiating the use of MLP with the peer.

To see an example of how to change the default endpoint discriminator, see the section "Changing the Default Endpoint Discriminator Example" at the end of this chapter.

Configuring MLP Interleaving and Queueing

Interleaving on MLP allows large packets to be multilink encapsulated and fragmented into a small enough size to satisfy the delay requirements of real-time traffic; small real-time packets are not multilink encapsulated and are sent between fragments of the large packets. The interleaving feature also provides a special transmit queue for the smaller, delay-sensitive packets, enabling them to be sent earlier than other flows.

Weighted fair queueing on MLP works on the packet level, not at the level of multilink fragments. Thus, if a small real-time packet gets queued behind a larger best-effort packet and no special queue has been reserved for real-time packets, the small packet will be scheduled for transmission only after all the fragments of the larger packet are scheduled for transmission.

Weighted fair queueing is now supported on all interfaces that support Multilink PPP, including MLP virtual access interfaces and virtual interface templates. Weighted fair-queueing is enabled by default.

Fair queueing on MLP overcomes a prior restriction. Previously, fair queueing was not allowed on virtual access interfaces and virtual interface templates. Interleaving provides the delay bounds for delay-sensitive voice packets on a slow link that is used for other best-effort traffic.

Interleaving applies only to interfaces that can configure a multilink bundle interface. These restrictions include virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces.

Multilink and fair queueing are not supported when a multilink bundle is off-loaded to a different system using Multichassis Multilink PPP (MMP). Thus, interleaving is not supported in MMP networking designs.

MLP support for interleaving can be configured on virtual templates, dialer interfaces, and ISDN BRI or PRI interfaces. To configure interleaving, complete the following tasks:

- Configure the dialer interface, BRI interface, PRI interface, or virtual template, as defined in the relevant chapters of this manual.
- Configure MLP and interleaving on the interface or template.

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

Configuring MLP Interleaving

To configure MLP and interleaving on a configured and operational interface or virtual interface template, use the following commands beginning in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 2	`Router(config-if)# ppp multilink interleave`	Enables interleaving of packets among the fragments of larger packets on an MLP bundle.
Step 3	`Router(config-if)# ppp multilink fragment delay` *milliseconds*	Specifies a maximum size, in units of time, for packet fragments on an MLP bundle.
Step 4	`Router(config-if)# ip rtp reserve` *lowest-udp-port range-of-ports* [*maximum-bandwidth*]	Reserves a special queue for real-time packet flows to specified destination UDP ports, allowing real-time traffic to have higher priority than other flows.
Step 5	`Router(config-if)# exit`	Exits interface configuration mode.
Step 6	`Router(config)# multilink virtual-template 1`	For virtual templates only, applies the virtual template to the multilink bundle.[1]

1. This step is not used for ISDN or dialer interfaces.

Interleaving statistics can be displayed by using the **show interfaces** command, specifying the particular interface on which interleaving is enabled. Interleaving data is displayed only if there are interleaves. For example, the following line shows interleaves:

```
Output queue: 315/64/164974/31191 (size/threshold/drops/interleaves)
```

Configuring MLP Inverse Multiplexer and Distributed MLP

The distributed MLP feature combines T1/E1 lines in a VIP on a Cisco 7500 series router into a bundle that has the combined bandwidth of the multiple T1/E1 lines. This is done using a VIP MLP link. You choose the number of bundles and the number of T1/E1 lines in each bundle, which allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line without having to purchase a T3 line.

Nondistributed MLP can only perform limited links, with CPU usage quickly reaching 90% with only a few T1/E1 lines running MLP. With distributed MLP, you can increase the router's total capacity.

The MLP Inverse Multiplexer feature was designed for Internet service providers (ISPs) that want to have the bandwidth of multiple T1 lines with performance comparable to that of an inverse multiplexer without the need of buying standalone inverse-multiplexing equipment. A Cisco router supporting VIPs can bundle multiple T1 lines in a CT3 or CE3 interface. Bundling is more economical than purchasing an inverse multiplexer, and eliminates the need to configure another piece of equipment.

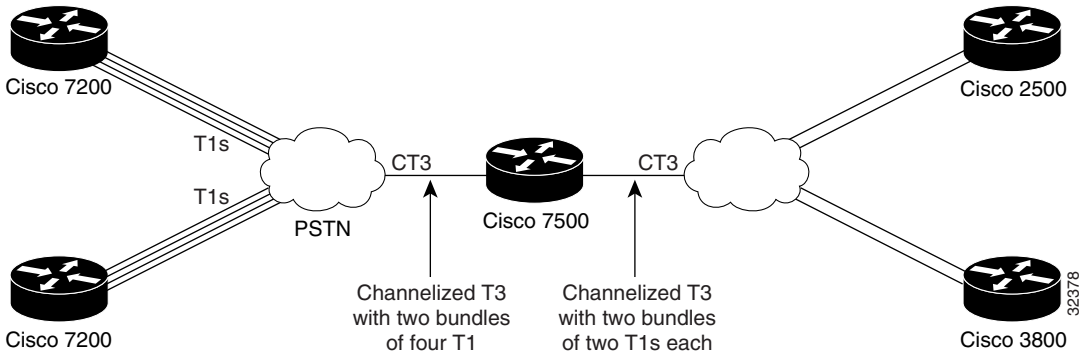
This feature supports the CT3 CE3 data rates without taxing the RSP and CPU by moving the data path to the VIP. This feature also allows remote sites to purchase multiple T1 lines instead of a T3 line, which is especially useful when the remote site does not need the bandwidth of an entire T3 line.

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

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

Figure 92 shows a typical network using a VIP MLP link. The Cisco 7500 series router is connected to the network with a CT3 line that has been configured with VIP MLP to carry two bundles of four T1 lines each. One of these bundles goes out to a Cisco 2500 series router and the other goes out to a Cisco 3800 series router.

Figure 92 Diagram of a Typical VIP MLP Topology



Before beginning the MLP Inverse Multiplexer configuration tasks, make note of the following prerequisites and restrictions.

Prerequisites

- Distributed CEF switching must be enabled for distributed MLP.
- One of the following port adapters is required:
  - CT3IP
  - PA-MC-T3
  - PA-MC-2T3+
  - PA-MC-E3
  - PA-MC-8T1
  - PA-MC-4T1
  - PA-MC-8E1
- All 16 E1s can be bundled from a PA-MC-E3 in a VIP4-80.

Restrictions

- The Multilink Inverse Multiplexer feature is supported only on the Cisco 7500 series routers.
- For bundles using IP, all lines in the bundle must have the same IP access list.
- Only one port adapter can be installed in a VIP.
- T1 and E1 lines cannot be mixed in a bundle.
- T1 lines in a bundle must have the same bandwidth.
- All lines in a bundle must have identical configurations.
- T1 lines can be combined in one bundle or up to 16 bundles per VIP.
- E1 lines can be combined in one bundle or up to 12 bundles per VIP.
- A maximum of eight T1 lines can be bundled on the VIP2-50 with two MB of SRAM.
- A maximum of 16 T1 lines can be bundled on the VIP2-50 with four or eight MB of SRAM.
- A maximum of 12 E1 lines can be bundled on the VIP2-50 with four or eight MB of SRAM.
- A maximum of 40 T1 lines can be bundled on the VIP4-80.
- Hardware compression is not supported.
- Encryption is not supported.
- Fancy/custom queueing is supported.
- MLP fragmentation is supported.
- Software compression is not recommended because CPU usage would negate performance gains.
- The maximum differential delay supported is 50 milliseconds.
- VIP CEF is limited to IP only; all other protocols are sent to the RSP.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation may result in better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation may be outweighed by the added load on the CPU.

To configure a multilink bundle, perform the tasks in the following sections:

- Enabling Distributed CEF Switching (Required for Distributed MLP)
- Creating a Multilink Bundle (Required)

- Assigning an Interface to a Multilink Bundle (Required)
- Disabling PPP Multilink Fragmentation (Optional)
- Verifying the MLP Inverse Multiplexer Configuration (Optional)

Enabling Distributed CEF Switching

To enable distributed MLP, first enable distributed CEF (dCEF) switching using the following command in global configuration mode:

Command	Purpose
`Router(config)# ip cef distributed`	Enables dCEF switching.

Creating a Multilink Bundle

To create a multilink bundle, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink` *group-number*	Assigns a multilink group number and begins interface configuration mode.
Step 2	`Router(config-if)# ip address` *address mask*	Assigns an IP address to the multilink interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp multilink`	Enables Multilink PPP.

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# no ip address`	Removes any specified IP address.
Step 2	`Router(config-if)# keepalive`	Sets the frequency of keepalive packets.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp multilink group` *group-number*	Restricts a physical link to joining only the designated multilink-group interface.
Step 5	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 6	`Router(config-if)# ppp authentication chap`	(Optional) Enables CHAP authentication.
Step 7	`Router(config-if)# pulse-time` *seconds*	(Optional) Configures DTR signal pulsing.

Disabling PPP Multilink Fragmentation

By default, PPP multilink fragmentation is enabled. To disable PPP multilink fragmentation, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)# ppp multilink fragment disable`	(Optional) Disables PPP multilink fragmentation.

Verifying the MLP Inverse Multiplexer Configuration

To display information about the newly created multilink bundle, use the **show ppp multilink** command in EXEC mode:

```
Router# show ppp multilink

Multilink1, bundle name is group1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links:4 active, 0 inactive (max not set, min not set)
 Serial1/0/0:1
 Serial1/0/0/:2
 Serial1/0/0/:3
 Serial1/0/0/:4
```

Monitoring and Maintaining PPP and MLP Interfaces

To monitor and maintain virtual interfaces, use the following command in EXEC mode:

Command	Purpose
`Router> show ppp multilink`	Displays MLP and MMP bundle information.

Configuration Examples for PPP and MLP

The following sections provide various PPP configuration examples:

- CHAP with an Encrypted Password Examples
- User Maximum Links Configuration Example
- MPPC Interface Configuration Examples
- IP Address Pooling Example
- DHCP Network Control Example
- PPP Reliable Link Examples
- MLP Examples
- MLP Interleaving and Queueing for Real-Time Traffic Example
- T3 Controller Configuration for an MLP Multilink Inverse Multiplexer Example

- Multilink Interface Configuration for Distributed MLP Example

CHAP with an Encrypted Password Examples

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

Configuration of Router yyy

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

Configuration of Router xxx

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

Configuration of Router zzz

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

When you look at the configuration file, the passwords will be encrypted and the display will look similar to the following:

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

User Maximum Links Configuration Example

The following example shows how to configure the username sTephen and establish a maximum of five connections. sTephen can connect through serial interface 1/0, which has a dialer map configured for it, or through PRI interface 0/0:23, which has dialer profile interface 0 dedicated to it.

The **aaa authorization network default local** command must be configured. PPP encapsulation and authentication must be enabled on all the interfaces that sTephen can connect to.

```
aaa new-model
aaa authorization network default local
enable secret saintstephen
enable password witharose
!
username sTephen user-maxlinks 5 password gardenhegoes
!
interface Serial0/0:23
```

```
 no ip address
 encapsulation ppp
 dialer pool-member 1
 ppp authentication chap
 ppp multilink
!
interface Serial1/0
 ip address 10.2.2.4 255.255.255.0
 encapsulation ppp
 dialer in-band
 dialer map ip 10.2.2.13 name sTephen 12345
 dialer-group 1
 ppp authentication chap
!
interface Dialer0
 ip address 10.1.1.4 255.255.255.0
 encapsulation ppp
 dialer remote-name sTephen
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp authentication chap
 ppp multilink
!
dialer-list 1 protocol ip permit
```

MPPC Interface Configuration Examples

The following example configures asynchronous interface 1 to implement MPPC and ignore the protocol field compression flag negotiated by LCP:

```
interface async1
 ip unnumbered ethernet0
 encapsulation ppp
 async default routing
 async dynamic routing
 async mode interactive
 peer default ip address 172.21.71.74
 compress mppc ignore-pfc
```

The following example creates a virtual access interface (virtual-template interface 1) and serial interface 0, which is configured for X.25 encapsulation. MPPC values are configured on the virtual-template interface and will ignore the negotiated protocol field compression flag.

```
interface ethernet0
 ip address 172.20.30.102 255.255.255.0
!
interface virtual-template1
 ip unnumbered ethernet0
 peer default ip address pool vtemp1
 compress mppc ignore-pfc
!
interface serial0
 no ipaddress
no ip mroute-cache
encapsulation x25
x25 win 7
x25 winout 7
x25 ips 512
x25 ops 512
clock rate 50000
!
```

```
ip local pool vtemp1 172.20.30.103 172.20.30.104
ip route 0.0.0.0 0.0.0.0 172.20.30.1
!
translate x25 31320000000000 virtual-template 1
```

IP Address Pooling Example

The following example configures a modem to dial in to a Cisco access server and obtain an IP address from the DHCP server. This configuration allows the user to log in and browse an NT network. Notice that the dialer 1 and group-async 1 interfaces are configured with the **ip unnumbered loopback** command, so that the broadcast can find the dialup clients and the client can see the NT network.

```
!
hostname secret
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default if-needed local
aaa authentication ppp chap local
enable secret 5 encrypted-secret
enable password EPassWd1
!
username User1 password 0 PassWd2
username User2 password 0 PassWd3
username User3 password 0 PassWd4
no ip domain-lookup
ip dhcp-server 10.47.0.131
async-bootp gateway 10.47.0.1
async-bootp nbns-server 10.47.0.131
isdn switch-type primary-4ess
!
!
controller t1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller t1 1
 framing esf
 clock source line secondary
 linecode b8zs
!
interface loopback 0
 ip address 10.47.252.254 255.255.252.0
!
interface ethernet 0
 ip address 10.47.0.5 255.255.252.0
 ip helper-address 10.47.0.131
 ip helper-address 10.47.0.255
 no ip route-cache
 no ip mroute-cache
!
interface serial 0
 no ip address
 no ip mroute-cache
 shutdown
!
interface serial 1
 no ip address
 shutdown
```

```
!
interface serial 0:23
 no ip address
 encapsulation ppp
 no ip mroute-cache
 dialer rotary-group 1
 dialer-group 1
 isdn incoming-voice modem
 no fair-queue
 no cdp enable
!
interface group-async 1
 ip unnumbered loopback 0
 ip helper-address 10.47.0.131
 ip tcp header-compression passive
 encapsulation ppp
 no ip route-cache
 no ip mroute-cache
 async mode interactive
 peer default ip address dhcp
 no fair-queue
 no cdp enable
 ppp authentication chap
 group-range 1 24
!
interface dialer 1
 ip unnumbered loopback 0
 encapsulation ppp
 dialer in-band
 dialer-group 1
 no peer default ip address
 no fair-queue
 no cdp enable
 ppp authentication chap
 ppp multilink
!
router ospf 172
 redistribute connected subnets
 redistribute static
 network 10.47.0.0 0.0.3.255 area 0
 network 10.47.156.0 0.0.3.255 area 0
 network 10.47.168.0 0.0.3.255 area 0
 network 10.47.252.0 0.0.3.255 area 0
!
ip local pool RemotePool 10.47.252.1 10.47.252.24
ip classless
ip route 10.0.140.0 255.255.255.0 10.59.254.254
ip route 10.2.140.0 255.255.255.0 10.59.254.254
ip route 10.40.0.0 255.255.0.0 10.59.254.254
ip route 10.59.254.0 255.255.255.0 10.59.254.254
ip route 172.23.0.0 255.255.0.0 10.59.254.254
ip route 192.168.0.0 255.255.0.0 10.59.254.254
ip ospf name-lookup
no logging buffered
access-list 101 deny ip any host 255.255.255.255
access-list 101 deny ospf any any
access-list 101 permit ip any any
dialer-list 1 protocol ip list 101
snmp-server community public RO
!
line con 0
line 1 24
 autoselect during-login
 autoselect ppp
```

```
 modem InOut
 transport input all
line aux 0
line vty 0 4
 password PassWd5
!
scheduler interval 100
end
```

DHCP Network Control Example

The following partial example adds the **ip dhcp-client network-discovery** command to the previous "IP Address Pooling Example" to allow peer routers to more dynamically discover DNS and NetBIOS name servers. If the **ip dhcp-client network-discovery** command is disabled, the system falls back to the static configurations made using the **async-bootp dns-server** and **async-bootp nb-server** global configuration commands.

```
!
hostname secret
!
aaa new-model
aaa authentication login default local
aaa authentication ppp default if-needed local
aaa authentication ppp chap local
enable secret 5 encrypted-secret
enable password EPassWd1
!
username User1 password 0 PassWd2
username User2 password 0 PassWd3
username User3 password 0 PassWd4
no ip domain-lookup
ip dhcp-server 10.47.0.131
ip dhcp-client network-discovery informs 2 discovers 2 period 12
async-bootp gateway 10.47.0.1
async-bootp nbns-server 10.47.0.131
isdn switch-type primary-4ess
.
.
.
```

PPP Reliable Link Examples

The following example enables PPP reliable link and STAC compression on BRI 0:

```
interface BRI0
 description Enables stac compression on BRI 0
 ip address 172.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 172.1.1.2 name baseball 14195386368
 compress stac
 ppp authentication chap
 dialer-group 1
 ppp reliable-link
```

The following example shows output of the **show interfaces** command when PPP reliable link is enabled. The LAPB output lines indicate that PPP reliable link is provided over LAPB.

```
Router# show interfaces serial 0
```

```
Serial0 is up, line protocol is up
  Hardware is HD64570
  Description: connects to enkidu s 0
  Internet address is 172.21.10.10/8
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set
  LCP Open
  Open: IPCP, CDP
  LAPB DTE, state CONNECT, modulo 8, k 7, N1 12048, N2 20
      T1 3000, T2 0, interface outage (partial T3) 0, T4 0, PPP over LAPB
      VS 1, VR 1, tx NR 1, Remote VR 1, Retransmissions 0
      Queues: U/S frames 0, I frames 0, unack. 0, reTx 0
      IFRAMEs 1017/1017 RNRs 0/0 REJs 0/0 SABM/Es 1/1 FRMRs 0/0 DISCs 0/0
  Last input 00:00:18, output 00:00:08, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0 (size/max/drops); Total output drops: 0
  Queueing strategy: weighted fair
  Output queue: 0/64/0 (size/threshold/drops)
     Conversations  0/1 (active/max active)
     Reserved Conversations 0/0 (allocated/max allocated)
  5 minute input rate 3000 bits/sec, 4 packets/sec
  5 minute output rate 3000 bits/sec, 7 packets/sec
     1365 packets input, 107665 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     2064 packets output, 109207 bytes, 0 underruns
     0 output errors, 0 collisions, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     4 carrier transitions
     DCD=up  DSR=up  DTR=up  RTS=up  CTS=up
```

MLP Examples

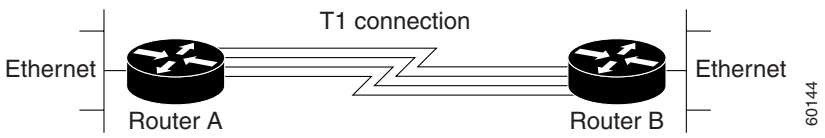
This section contains the following MLP examples:

- MLP on Synchronous Serial Interfaces Example
- MLP on One ISDN BRI Interface Example
- MLP on Multiple ISDN BRI Interfaces Example
- MLP Using Multilink Group Interfaces over ATM Example
- Changing the Default Endpoint Discriminator Example

MLP on Synchronous Serial Interfaces Example

MLP provides characteristics most similar to hardware inverse multiplexers, with good manageability and Layer 3 services support. Figure 93 shows a typical inverse multiplexing application using two Cisco routers and Multilink PPP over four T1 lines.

Figure 93 Inverse Multiplexing Application Using Multilink PPP



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

Router A Configuration

```
hostname RouterA
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.1.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

Router B Configuration

```
hostname RouterB
!
!
username RouterB password your_password
ip subnet-zero
multilink virtual-template 1
!
interface Virtual-Template1
 ip unnumbered Ethernet0
```

```
 ppp authentication chap
 ppp multilink
!
interface Serial0
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial1
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial2
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Serial3
 no ip address
 encapsulation ppp
 no fair-queue
 ppp multilink
 pulse-time 3
!
interface Ethernet0
 ip address 10.17.2.254 255.255.255.0
!
router rip
network 10.0.0.0
!
end
```

MLP on One ISDN BRI Interface Example

The following example enables MLP on BRI interface 0. Because an ISDN interface is a rotary group by default, when one BRI is configured, no dialer rotary group configuration is required.

```
interface bri 0
 description connected to ntt 81012345678902
 ip address 172.31.1.7 255.255.255.0
 encapsulation ppp
 dialer idle-timeout 30
 dialer load-threshold 40 either
 dialer map ip 172.31.1.8 name atlanta 81012345678901
 dialer-group 1
 ppp authentication pap
 ppp multilink
```

MLP on Multiple ISDN BRI Interfaces Example

The following example configures multiple ISDN BRI interfaces to belong to the same dialer rotary group for Multilink PPP. The **dialer rotary-group** command is used to assign each of the ISDN BRI interfaces to that dialer rotary group.

```
interface BRI0
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI1
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface BRI2
 no ip address
 encapsulation ppp
 dialer idle-timeout 500
 dialer rotary-group 0
 dialer load-threshold 30 either
!
interface Dialer0
 ip address 10.0.0.2 255.0.0.0
 encapsulation ppp
 dialer in-band
 dialer idle-timeout 500
 dialer map ip 10.0.0.1 name atlanta broadcast 81012345678901
 dialer load-threshold 30 either
 dialer-group 1
 ppp authentication chap
 ppp multilink
```

MLP Using Multilink Group Interfaces over ATM Example

The following example configures MLP over an ATM PVC using a multilink group:

```
interface multilink 1
 ip address 10.200.83.106 255.255.255.252
 ip tcp header-compression iphc-format delay 20000
 service policy output xyz
 encapsulation ppp
 ppp multilink
 ppp multilink fragment delay 10
 ppp multilink interleave
 ppp timeout multilink link remove 10
 ip rtp header-compression iphc-format

interface virtual-template 3
 bandwidth 128
 ppp multilink group 1

interface atm 4/0.1 point-to-point
 pvc 0/32
 abr 100 80
 protocol ppp virtual-template 3
```

Changing the Default Endpoint Discriminator Example

The following partial example changes the MLP endpoint discriminator from the default CHAP host name C-host1 to the E.164-compliant telephone number `1 603 555-1212`:

.

```
.
.
interface dialer 0
 ip address 10.1.1.4 255.255.255.0
 encapsulation ppp
 dialer remote-name R-host1
 dialer string 23456
 dialer pool 1
 dialer-group 1
 ppp chap hostname C-host1
 ppp multilink endpoint phone 16035551212
.
.
.
```

MLP Interleaving and Queueing for Real-Time Traffic Example

The following example defines a virtual interface template that enables MLP interleaving and a maximum real-time traffic delay of 20 milliseconds, and then applies that virtual template to the MLP bundle:

```
interface virtual-template 1
 ip unnumbered ethernet 0
 ppp multilink
 ppp multilink interleave
 ppp multilink fragment delay 20
 ip rtp interleave 32768 20 1000
multilink virtual-template 1
```

The following example enables MLP interleaving on a dialer interface that controls a rotary group of BRI interfaces. This configuration permits IP packets to trigger calls.

```
interface BRI 0
 description connected into a rotary group
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 1
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 2
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 3
 no ip address
 encapsulation ppp
 dialer rotary-group 1
!
interface BRI 4
 encapsulation ppp
 dialer rotary-group 1
!
interface Dialer 0
 description Dialer group controlling the BRIs
 ip address 10.1.1.1 255.255.255.0
 encapsulation ppp
 dialer map ip 10.1.1.2 name angus 14802616900
 dialer-group 1
 ppp authentication chap
```

```
! Enables Multilink PPP interleaving on the dialer interface and reserves
! a special queue.
 ppp multilink
 ppp multilink interleave
 ip rtp reserve 32768 20 1000
! Keeps fragments of large packets small enough to ensure delay of 20 ms or less.
 ppp multilink fragment delay 20
dialer-list 1 protocol ip permit
```

T3 Controller Configuration for an MLP Multilink Inverse Multiplexer Example

In the following example, the T3 controller is configured and four channelized interfaces are created:

```
controller T3 1/0/0
framing m23
cablelength 10
t1 1 timeslots 1-24
t1 2 timeslots 1-24
t1 3 timeslots 1-24
t1 4 timeslots 1-24
```

Multilink Interface Configuration for Distributed MLP Example

In the following example, four multilink interfaces are created with distributed CEF switching and MLP enabled. Each of the newly created interfaces is added to a multilink bundle.

```
interface multilink1
 ip address 10.0.0.0 10.255.255.255
 ppp chap hosstname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:1
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:2
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:3
 no ip address
 encapsulation ppp
 ip route-cache distributed
 no keepalive
 ppp chap hostname group 1
 ppp multilink
 ppp multilink group 1

interface serial 1/0/0/:4
 no ip address
```

```
encapsulation ppp
ip route-cache distributed
no keepalive
ppp chap hostname group 1
ppp multilink
ppp multilink group 1
```

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 Multiprotocol Label Switching Command Reference* at http://www.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.

- **encapsulation ppp**
- **ppp authentication**
- **aaa authentication ppp**
- **ppp use-tacacs**
- **ppp quality**
- **ppp multilink**
- **ppp multilink interleave**
- **ppp multilink fragment delay**
- **multilink virtual-template**
- **compress**
- **compress stac**
- **ip address-pool**
- **ip dhcp-server**

- **ip local pool**
- **ip dhcp-client**
- **interface**
- **ppp reliable-link**
- **ppp bridge appletalk**
- **dialer idle-timeout**
- **dialer in-band**
- **interface bri**
- **dialer rotary-group**
- **dialer load-threshold**
- **protocol ppp**
- **ppp multilink endpoint**

Feature Information for Media-Independent PPP and Multilink PPP

Table 36 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, Catalyst OS, and Cisco IOS XE 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 36 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 36 Feature Information for Media-Independent PPP and Multilink PPP

Feature Name	Releases	Feature Information
Media-Independent PPP and Multilink PPP	Cisco IOS XE Release 2.1 12.2(33)SXI	This feature was introduced on Cisco ASR 1000 Series Routers. This feature was integrated into Cisco IOS Release 12.2(33)SXI.

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.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

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 267.

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 269.

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**

Multilink PPP Minimum Links Mandatory

First Published: 12.1(11b)E
Last Updated: February 28, 2006

Multilink PPP allows multiple PPP links to be established in parallel to the same destination. Multilink PPP is often used with dialup lines or ISDN connections to easily increase the amount of bandwidth between points.

With the introduction of the Multilink PPP Minimum Links Mandatory feature, you can configure the minimum number of links in a Multilink PPP (MLP) bundle required to keep that bundle active by entering the **ppp multilink min-links** *links* **mandatory** command. When you configure this command, all Network Control Protocols (NCPs) for an MLP bundle are disabled until the MLP bundle has the required minimum number of links. When a new link is added to the MLP bundle that brings the number of links up to the required minimum number of links, the NCPs are activated for the MLP bundle. When a link is removed from an MLP bundle, and the number of links falls below the required minimum number of links for that MLP bundle, the NCPs are disabled for that MLP bundle.

History for the Multilink PPP Minimum Links Mandatory Feature

Release	Modification
12.1(11b)E	This feature was introduced.
12.2(13)T	This feature was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.2(15)B	This feature was integrated into Cisco IOS Release 12.2(15)B and support for the Cisco 7401ASR and the Cisco 6400 series was added.
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This documentation describes the Multilink PPP Minimum Links Mandatory feature for Cisco IOS Releases 12.2(13)T, 12.2(14)S, and 12.2(15)B.

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

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

Information About Multilink PPP Minimum Links Mandatory

You must understand the following concepts to configure this feature:

PPP Encapsulation Overview

PPP, described in RFC 1661, encapsulates network layer protocol information over point-to-point links. You can configure PPP on the following types of physical interfaces:

- Asynchronous serial
- High-Speed Serial Interface (HSSI)
- ISDN
- Synchronous serial

When PPP encapsulation is enabled on physical interfaces, PPP can also be in effect on calls placed by the dialer interfaces that use the physical interfaces.

PPP supports option 3, authentication using Challenge Handshake Authentication Protocol (CHAP) or Password Authentication Protocol (PAP); option 4, Link Quality Monitoring (LQM); and option 5, Magic Number configuration options. Cisco IOS software always sends option 5 and negotiates for options 3 and 4 if so configured. All other options are rejected.

Magic Number support is available on all serial interfaces. PPP always attempts to negotiate for Magic Numbers, which are used to detect looped-back lines. Depending on how the down-when-looped command is configured, the router might shut down a link if it detects a loop.

Cisco IOS software provides the CHAP and PAP on serial interfaces running PPP encapsulation. For detailed information about authentication, refer to the *Cisco IOS Security Configuration Guide*.

Multilink PPP Overview

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.

How to Configure Multilink PPP Minimum Links Mandatory

This section contains the following procedures:

- Configuring PPP, page 3 (required)
- Configuring Multilink PPP, page 5 (required)
- Configuring Multilink PPP Minimum Links Mandatory, page 7 (required)

Configuring PPP

Perform this task to configure PPP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type slot/port*
4. **encapsulation ppp**
5. **ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [**list-name** | **default**] [**callin**]
6. **exit**
7. **username** *name* **password** *secret*

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 slot*/*port* **Example:** `Router(config)# interface serial 1/0`	Configures an interface and enters interface configuration mode. • *type*—Type of interface to be configured. • *slot*—Number of the slot being configured. • *port*—Number of the port being configured. **Note** Refer to the appropriate hardware manual for slot and port information.
Step 4	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.

	Command or Action	Purpose
Step 5	**ppp authentication** {**chap** | **chap pap** | **pap chap** | **pap**} [**if-needed**] [*list-name* | **default**] [**callin**] **Example:** `Router(config-if)# ppp authentication chap`	(Optional) Defines the authentication methods supported and the order in which they are used. • **chap**—Enables CHAP on a serial interface. • **chap pap**—Enables CHAP and PAP on a serial interface and configures CHAP to be used first. • **pap chap**—Enables CHAP and PAP on a serial interface and configures PAP to be used first. • **pap**—Enables PAP on a serial interface. • **if-needed**—(Optional) Used with TACACS and extended TACACS. Does not perform CHAP or PAP authentication if the user has already provided authentication. This option is available only on asynchronous interfaces. • *list-name*—(Optional) Used with authentication, authorization, and accounting (AAA). Specifies the name of a list of methods of authentication to use. If no list name is specified, the system uses the default. The list is created with the **aaa authentication ppp** command. • **default**—(Optional) The name of the method list is created with the **aaa authentication ppp** command. • **callin**—(Optional) Specifies authentication on incoming (received) calls only.
Step 6	**exit** **Example:** `Router(config-if)# exit`	Exits interface configuration mode.
Step 7	**username** *name* **password** *secret* **Example:** `Router(config)# username username1 password password1`	(Optional) Specifies the password to be used in CHAP or PAP caller identification. • *name*—Assigns a host name, server name, user ID, or command name. The *name* argument can be only a single word and not more than one word. Blank spaces and quotation marks are not allowed. • *secret*—Specifies the secret for the local router or the remote device. The secret is encrypted when it is stored on the local router. The secret can consist of any string of up to 11 ASCII characters. There is no limit to the number of username and password combinations that can be specified, allowing any number of remote devices to be authenticated.

Configuring Multilink PPP

Perform this task to configure MLP.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface multilink** *group-number*
4. **ip address** *address*
5. **encapsulation ppp**
6. **ppp multilink**
7. **ppp multilink max-links** *links*
8. **ppp multilink min-links** *links*
9. **bridge-group** *bridge-group-number*
10. **no shutdown**

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 multilink** *group-number* **Example:** `Router(config)# interface multilink 3`	Creates a multilink bundle and enters interface configuration mode. • *group-number*—Specifies the number of the multilink bundle. Valid range is from 0 to 214748364.
Step 4	**ip address** *address* **Example:** `Router(config-if)# ip address 172.16.0.0`	Assigns an IP address to the interface.
Step 5	**encapsulation ppp** **Example:** `Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 6	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP.

	Command or Action	Purpose
Step 7	**ppp multilink max-links** *links* **Example:** `Router(config-if)# ppp multilink max-links 100`	(Optional) Limits the maximum number of links that MLP can dial for dynamic allocation. • *links*—Maximum number of links, in the range 0 to 255.
Step 8	**ppp multilink min-links** *links* **Example:** `Router(config-if)# ppp multilink min-links 5`	(Optional) Specifies the preferred minimum number of links in an MLP bundle. • *links*—Minimum number of links, in the range from 0 to 255.
Step 9	**bridge-group** *bridge-group-number* **Example:** `Router(config-if)# bridge-group 2`	(Optional) Specifies the bridge group to which this interface belongs. • *bridge-group-number*—Number of the bridge group to which the interface belongs. Valid values are from 1 to 255. **Note** Use this command only if bridging is enabled for this interface.
Step 10	**no shutdown** **Example:** `Router(config-if)# no shutdown`	Enables the interface.

Configuring Multilink PPP Minimum Links Mandatory

Perform this task to configure the minimum number of links in an MLP bundle required to keep that bundle active.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ppp multilink**
4. **ppp multilink min-links** *links* **mandatory**

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	**ppp multilink** **Example:** `Router(config-if)# ppp multilink`	Enables MLP.
Step 4	**ppp multilink min-links** *links* **mandatory** **Example:** `Router(config-if)# ppp multilink min-links 5 mandatory`	Specifies the required minimum number of links in a Multilink PPP (MLP) bundle. • If the minimum number of links in the MLP bundle falls below the number specified by the *links* argument, the MLP bundle is disabled. • *links*—Minimum number of links, in the range from 0 to 255.

Verifying the Multilink PPP Minimum Links Mandatory Configuration

Perform this task to verify configuration of the Multilink PPP Minimum Links Mandatory feature.

SUMMARY STEPS

1. **enable**
2. **show running-config** [**interface** *type number*] [**linenum**]
3. **show interfaces multilink** *group-number*
4. **show ppp multilink**
5. **show interfaces multilink** *group-number* **stat**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show running-config** [**interface** *type number*] [**linenum**] **Example:** `Router# show running-config`	Displays the contents of the currently running configuration file or the configuration for a specific interface, or map class information.
Step 3	**show interfaces multilink** *group-number* **Example:** `Router# show interfaces multilink 3`	(Optional) Displays statistics for the specified multilink bundle.

	Command or Action	Purpose
Step 4	**show ppp multilink** **Example:** `Router# show ppp multilink`	(Optional) Displays information about all existing multilink bundles and their member links.
Step 5	**show interfaces multilink** *group-number* **stat** **Example:** `Router# show interfaces multilink 3 stat`	(Optional) Displays traffic statistics for a multilink bundle.

Examples

The following is sample output from the **show running-config** command that shows that the Multilink PPP Minimum Links Mandatory feature is configured on interface bri0:

```
Router# show running-config
.
.
.
interface multilink1
 ip address 10.0.0.0 255.255.255.0
   encapsulation ppp
   ppp authentication chap
   ppp multilink
   ppp multilink max-links 100
   ppp multilink min-links 10 mandatory

  interface BRI2/1
   no ip address
   encapsulation ppp
   dialer pool-member 2
   no fair-queue
   no cdp enable
   ppp authentication chap
   ppp multilink

 interface bri 0
  description connected to abc 81012345678902
  ip address 172.16.0.10 255.255.255.0
  encapsulation ppp
  dialer idle-timeout 30
  dialer map ip 172.16.0.0 name cisco 81012345678901
  dialer-group 1
  ppp authentication pap
  ppp multilink
  ppp multilink min-links 2 mandatory ! Indicates that the Multilink PPP Minimum Links
                                        Mandatory feature is enabled.
```

Configuration Examples for Multilink PPP Minimum Links Mandatory

This section provides the following configuration examples:

- Configuring PPP: Example, page 10

Configuring PPP: Example

The following example shows how to configure PPP on a serial interface with CHAP authentication:

```
interface serial 1/0
 encapsulation ppp
 ppp authentication chap
 exit
username abc password password1
```

Configuring Multilink PPP: Example

The following example shows how to configure MLP. In this example, the MLP bundle is configured with CHAP authentication. The minimum number of links for this MLP bundle is 5 and the maximum number of links is 100.

```
interface multilink 3
 ip address 172.16.0.0
 encapsulation ppp
 ppp multilink
 ppp multilink max-links 100
 ppp multilink min-links 5
 bridge-group 2
 no shutdown
```

Configuring Multilink PPP Minimum Links Mandatory: Example

The following example shows how to configure an MLP bundle to be required to have at least five active sessions:

```
ppp multilink
ppp multilink min-links 5 mandatory
```

Additional References

The following sections provide references related to the Multilink PPP Minimum Links Mandatory feature.

Related Documents

Related Topic	Document Title
Dial technologies commands	*Cisco IOS Dial Technologies Command Reference*
PPP and multilink PPP configuration	• *Configuring Media-Independent PPP and Multilink PPP* • *Configuring PPP and Multilink PPP* • *Criteria for Naming Multilink PPP Bundles* • *Multichassis Multilink PPP (MMP)* • *Router-to-Router Async Multilink PPP* • *Troubleshooting Async Multilink PPP Operations*

Standards

Standards	Title
No new or modified standards are supported by this feature and support for existing standards has not been modified by this feature.	—

MIBs

MIBs	MIBs Link
No new or modified MIBs are supported by this feature and support for existing standards has not been modified 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

RFCs

RFCs	Title
RFC 1990	*The PPP Multilink Protocol (MP)*

Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of 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/dia/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*.

- **multilink min-links**
- **ppp multilink links minimum**

Callback and Bandwidth Allocation Configuration

Configuring PPP Callback

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes how to configure PPP callback for dial-on-demand routing (DDR). It includes the following main sections:

- PPP Callback for DDR Overview
- How to Configure PPP Callback for DDR
- MS Callback Overview
- How to Configure MS Callback
- Configuration Examples for PPP Callback

This feature implements the following callback specifications of RFC 1570:

- For the client—Option 0, location is determined by user authentication.
- For the server—Option 0, location is determined by user authentication; Option 1, dialing string; and Option 3, E.164 number.

Return calls are made through the same dialer rotary group but not necessarily the same line as the initial call.

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

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

PPP Callback for DDR Overview

PPP callback provides a client/server relationship between the endpoints of a point-to-point connection. PPP callback allows a router to request that a dialup peer router call back. The callback feature can be used to control access and toll costs between the routers.

When PPP callback is configured on the participating routers, the calling router (the callback client) passes authentication information to the remote router (the callback server), which uses the host name and dial string authentication information to determine whether to place a return call. If the authentication is successful, the callback server disconnects and then places a return call. The remote username of the return call is used to associate it with the initial call so that packets can be sent.

Both routers on a point-to-point link must be configured for PPP callback; one must function as a callback client and one must be configured as a callback server. The callback client must be configured to initiate PPP callback requests, and the callback server must be configured to accept PPP callback requests and place return calls.

See the section "MS Callback Overview" later in this chapter if you are using PPP callback between a Cisco router or access server and client devices configured for Windows 95 and Windows NT.

Note If the return call fails (because the line is not answered or the line is busy), no retry occurs. If the callback server has no interface available when attempting the return call, it does not retry.

How to Configure PPP Callback for DDR

To configure PPP callback for DDR, perform the following tasks:

- Configuring a Router As a Callback Client (Required)
- Configuring a Router As a Callback Server (Required)

For an example of configuring PPP callback, see the section "Configuration Examples for PPP Callback" at the end of this chapter.

Configuring a Router As a Callback Client

To configure a router interface as a callback client, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface type number`	Specifies the interface and enters interface configuration mode.
Step 2	`Router(config-if)# dialer in-band [no-parity | odd-parity]`	Enables DDR. Specifies parity, if needed, on synchronous or asynchronous serial interfaces.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp authentication {chap | pap}`	Enables CHAP or PAP authentication.
Step 5	`Router(config-if)# dialer map protocol next-hop-address name hostname dial-string`	Maps the next hop address to the host name and phone number.

	Command	Purpose
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 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
```

Dial Access Specialized Features

Configuring per-User Configuration

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.
Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Routers.

This chapter describes per-user configuration, a large-scale dial solution. It includes the following main sections:

- Per-User Configuration Overview
- How to Configure a AAA Server for Per-User Configuration
- Monitoring and Debugging Per-User Configuration Settings
- Configuration Examples for Per-User Configuration

This set of features is supported on all platforms that support Multilink PPP (MLP).

A virtual access interface created dynamically for any user dial-in session is deleted when the session ends. The resources used during the session are returned for other dial-in uses.

When a specific user dials in to a router, the use of a per-user configuration from an authentication, authorization, and accounting (AAA) server requires that AAA is configured on the router and that a configuration for that user exists on the AAA server.

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

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

Per-User Configuration Overview

Per-user configuration provides a flexible, scalable, easily maintained solution for customers with a large number of dial-in users. This solution can tie together the following dial-in features:

- Virtual template interfaces, generic interface configuration and router-specific configuration information stored in the form of a virtual template interface that can be applied (*cloned*) to a virtual access interface each time any user dials in. This configuration is described in the chapter "Configuring Virtual Template Interfaces" in this publication.
- AAA per-user security and interface configuration information stored on a separate AAA server and sent by the AAA server to the access server or router in response to authorization requests during the PPP authentication phase. The per-user configuration information can add to or override the generic configuration on a virtual interface.
- Virtual profiles, which can use either or both of the two sources of information listed in the previous bullets for virtual interface configuration. When a user dials in, virtual profiles can apply the generic interface configuration and then apply the per-user configuration to create a unique virtual access interface for that user. This configuration is described in the chapter "Configuring Virtual Profiles" in this publication.

The per-user configuration feature provides these benefits:

- Maintenance ease for service providers with a large number of access servers and a very large number of dial-in users. Service providers need not update all their routers and access servers when user-specific information changes; instead, they can update one AAA server.
- Scalability. By separating generic virtual interface configuration on the router from the configuration for each individual, Internet service providers and other enterprises with large numbers of dial-in users can provide a uniquely configured interface for each individual user. In addition, by separating the generic virtual interface configuration from the physical interfaces on the router, the number and types of physical interfaces on the router or access server are not intrinsic barriers to growth.

General Operational Processes

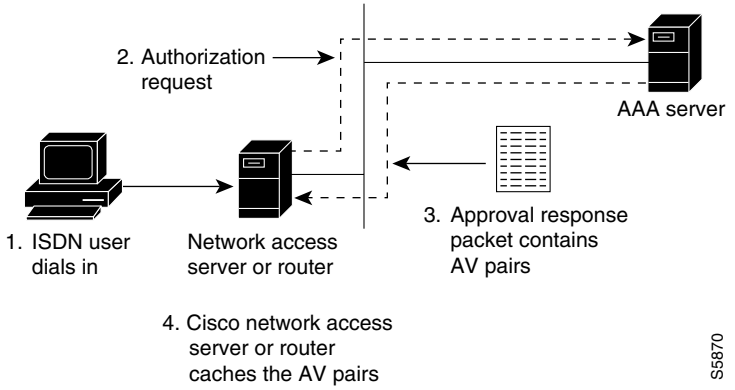
In general, the per-user configuration process on the Cisco router or network access server proceeds as follows:

1. The user dials in.
2. The authentication and authorization phases occur.
   a. If AAA is configured, the router sends an authorization request to the AAA server.
   b. If the AAA server has information (attribute-value or AV pairs, or other configuration parameters) that defines a configuration for the specific user, the server includes it in the information in the approval response packet.

      Figure 34 illustrates the request and response part of the process that happens when a user dials in, given that AAA is configured and that the AAA server has per-user configuration information for the dial-in user.
   c. The router looks for AV pairs in the AAA approval response.
   d. The router caches the configuration parameters.

Note TACACS servers treat authentication and authorization as two phases; RADIUS servers combine authentication and authorization into a single step. For more detailed information, refer to your server documentation.

Figure 34 Per-User Configuration Authentication and Authorization



3. A virtual access interface is created for this user.
   a. The router finds the virtual template that is set up for virtual profiles, if any, and applies the commands to the virtual access interface.
   b. The router looks for the AV pairs to apply to this virtual access interface to configure it for the dial-in user.
   c. The AV pairs are sent to the Cisco IOS command-line parser, which interprets them as configuration commands and applies them to configure this virtual access interface.

The result of this process is a virtual access interface configured uniquely for the dial-in user.

When the user ends the call, the virtual access interface is deleted and its resources are returned for other dial-in uses.

Note The use of virtual profiles can modify the process that occurs between the user dial-in and the use of AAA configuration information. For more information, see the chapter "Configuring Virtual Profiles" in this publication.

Operational Processes with IP Address Pooling

During IP Control Protocol (IPCP) address negotiation, if an IP pool name is specified for a user, the network access server checks whether the named pool is defined locally. If it is, no special action is required and the pool is consulted for an IP address.

If the required pool is not present (either in the local configuration or as a result of a previous download operation), an authorization call to obtain it is made using the special username:

`pools-`*`nas-name`*

where *nas-name* is the configured name of the network access server. In response, the AAA server downloads the configuration of the required pool.

This pool username can be changed using Cisco IOS configuration, for example:

```
aaa configuration config-name nas1-pools-definition.cisco.us
```

This command has the effect of changing the username that is used to download the pool definitions from the default name "pools-*nas-name*" to "nas1-pools-definition.cisco.com."

On a TACACS+ server, the entries for an IP address pool and a user of the pool might be as follows:

```
user = nas1-pools {
     service = ppp protocol = ip {
        pool-def#1 = "aaa 10.0.0.1 10.0.0.3"
        pool-def#2 = "bbb 10.1.0.1 10.1.0.10"
        pool-def#3 = "ccc 10.2.0.1 10.2.0.20"
        pool-timeout=60
    }
}

user = georgia {
    login = cleartext lab
    service = ppp protocol = ip {
                  addr-pool=bbb
    }
}
```

On a RADIUS server, the entries for the same IP address pool and user would be as follows:

```
nas1-pools      Password = "cisco" User-Service-Type=Outbound-User
        cisco-avpair = "ip:pool-def#1=aaa 10.0.0.1 10.0.0.3",
        cisco-avpair = "ip:pool-def#2=bbb 10.1.0.1 10.1.0.10",
        cisco-avpair = "ip:pool-def#3=ccc 10.2.0.1 10.2.0.20",
        cisco-avpair = "ip:pool-timeout=60"

georgia Password = "lab"
        User-Service-Type = Framed-User,
        Framed-Protocol = PPP,
                cisco-avpair = "ip:addr-pool=bbb"
```

Note This entry specifies a User-Service-Type of Outbound-User. This attribute is supplied by the network access server to prevent ordinary logins from using the well-known username and password combination of nas1-pools/cisco.

Pools downloaded to a Cisco network access server are not retained in nonvolatile memory and automatically disappear whenever the access server or router restarts. Downloaded pools can also be made to time out automatically by adding a suitable AV pair. For more information, see the section "Supported Attrubutes for AV Pairs" and the pool-timeout attribute in Table 5. Downloaded pools are marked as *dynamic* in the output of the **show ip local pool** command.

Deleting Downloaded Pools

To delete downloaded pools, you can do either of the following:

- Manually delete the definition from the network access server. For example, if "bbb" is the name of a downloaded pool, you can enter the Cisco IOS **no ip local pool bbb** command.

  Deleting a pool definition does not interrupt service for current users. If a pool is deleted and then redefined to include a pool address that is currently allocated, the new pool understands and tracks the address as expected.

- Set an AV pair pool-timeout value; this is a more desirable solution.

The pool-timeout AV pair starts a timer when the pool is downloaded. Once the timer expires, the pools are deleted. The next reference to the pools again causes an authorization call to be made, and the pool definition is downloaded again. This method allows definitions to be made and changed on the AAA server and propagated to network access servers.

Supported Attributes for AV Pairs

Table 5 provides a partial list of the Cisco-specific supported attributes for AV pairs that can be used for per-user virtual interface configuration. For complete lists of Cisco-specific, vendor-specific, and TACACS+ supported attributes, see the *Cisco IOS Security Configuration Guide* and *Cisco IOS Security Command Reference*.

Table 5 Partial List of Cisco-Specific Supported AV Pair Attributes

Attribute	Meaning
inacl#	An input access list definition. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For Internet Protocol Exchange (IPX), only extended syntax is recognized. The value of this attribute is the text that comprises the body of a named access list definition.
outacl#[1]	An output access list definition. For IP, standard or extended access list syntax can be used. For IPX, only extended syntax is recognized. The value of this attribute is the text that comprises the body of a named access list definition.
rte-fltr-in#	An input route filter. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For IPX, only extended syntax is recognized. The first line of this filter must specify a routing process. Subsequent lines comprise the body of a named access list.
rte-fltr-out#	An output route filter. For IP, standard or extended access list syntax can be used, although you cannot mix them within a single list. For IPX, only extended syntax is recognized. The first line of this filter must specify a routing process. Subsequent lines comprise the body of a named access list.
route#[2]	Static routes, for IP and IPX. The value is text of the form *destination-address mask* [*gateway*].
sap#	IPX static Service Advertising Protocol (SAP). The value is text from the body of an **ipx sap** configuration command.
sap-fltr-in#	IPX input SAP filter. Only extended access list syntax is recognized. The value is text from the body of an extended IPX **access-list** configuration command. (The Novell socket number for SAP filtering is 452.)
sap-fltr-out#	IPX output SAP filter. Only extended **access-list** command syntax is recognized. The value is text from the body of an extended IPX access-list configuration command.
pool-def#	An IP pool definition. The value is text from the body of an **ip local pool** configuration command.
pool-timeout	An IP pool definition. The body is an integer representing a timeout, in minutes.

1. The “outacl” attribute still exists and retains its old meaning.
2. The “route” attribute, without a trailing #, is still recognized for backward compatibility with the TACACS+ protocol specification, but if multiple static routes are required in TACACS+, full “route#” names will need to be employed.

Table 6 provides examples for each attribute on an AAA TACACS+ server.

Table 6 TACACS+ Server AV Pair Examples for Each Attribute

Attribute	TACACS+ Server Examples

inacl#	**IP:**
	`inacl#3="permit ip any any precedence immediate"`
	`inacl#4="deny igrp 10.0.1.2 255.255.0.0 any"`
	IPX:
	`inacl#1="deny 3C01.0000.0000.0001"`
	`inacl#2="deny 4C01.0000.0000.0002"`

outacl#	`outacl#2="permit ip any any precedence immediate"` `outacl#3="deny igrp 10.0.9.10 255.255.0.0 any"`

rte-fltr-in#	**IP:**
	`rte-fltr-in#1="router igrp 60"`
	`rte-fltr-in#3="permit 10.0.3.4 255.255.0.0"`
	`rte-fltr-in#4="deny any"`
	IPX:
	`rte-fltr-in#1="deny 3C01.0000.0000.0001"`
	`rte-fltr-in#2="deny 4C01.0000.0000.0002"`

rte-fltr-out#	`rte-fltr-out#1="router igrp 60"`
	`rte-fltr-out#3="permit 10.0.5.6 255.255.0.0"`
	`rte-fltr-out#4="permit any"`

route#	**IP:**
	`route#1="10.0.0.0 255.0.0.0 1.2.3.4"`
	`route#2="10.1.0.0 255.0.0.0"`
	IPX:
	`route#1="4C000000 ff000000 10.12.3.4"`
	`route#2="5C000000 ff000000 10.12.3.5"`

sap#	`sap#1="4 CE1-LAB 1234.0000.0000.0001 451 4"` `sap#2="5 CE3-LAB 2345.0000.0000.0001 452 5"`
sap-fltr-in#	`sap-fltr-in#1="deny 6C01.0000.0000.0001"` `sap-fltr-in#2="permit -1"`
sap-fltr-out#	`sap-fltr-out#1="deny 6C01.0000.0000.0001"` `sap-fltr-out#2="permit -1"`

pool-def#	`pool-def#1 = "aaa 10.0.0.1 1.0.0.3"`
	`pool-def#2 = "bbb 10.1.0.1 2.0.0.10"`
	`pool-def#3 = "ccc 10.2.0.1 3.0.0.20"`

pool-timeout	`pool-timeout=60`

Table 7 provides examples for each attribute on an AAA RADIUS server.

Table 7 RADIUS Server AV Pair Examples for Each Attribute

Attribute	RADIUS Server Examples
lcp:interface-config[1]	`cisco-avpair = "lcp:interface-config=ip address 10.0.0.0 255.255.255.0",`
inacl#	`cisco-avpair = "ip:inacl#3=permit ip any any precedence immediate",` `cisco-avpair = "ip:inacl#4=deny igrp 10.0.1.2 255.255.0.0 any",`

Table 7 RADIUS Server AV Pair Examples for Each Attribute (continued)

Attribute	RADIUS Server Examples
outacl#	`cisco-avpair = "ip:outacl#2=permit ip any any precedence immediate",` `cisco-avpair = "ip:outacl#3=deny igrp 10.0.9.10 255.255.0.0 any",`

rte-fltr-in#	**IP:**
	`cisco-avpair = "ip:rte-fltr-in#1=router igrp 60",`
	`cisco-avpair = "ip:rte-fltr-in#3=permit 10.0.3.4 255.255.0.0",`
	`cisco-avpair = "ip:rte-fltr-in#4=deny any",`
	IPX:
	`cisco-avpair = "ipx:rte-fltr-in=deny 3C01.0000.0000.0001",`

rte-fltr-out#	`cisco-avpair = "ip:rte-fltr-out#1=router igrp 60",`
	`cisco-avpair = "ip:rte-fltr-out#3=permit 10.0.5.6 255.255.0.0",`
	`cisco-avpair = "ip:rte-fltr-out#4=permit any",`

route#	**IP:**
	`cisco-avpair = "ip:route=3.10.0.0 255.0.0.0 1.2.3.4",`
	`cisco-avpair = "ip:route=4.10.0.0 255.0.0.0",`
	IPX:
	`cisco-avpair = "ipx:route=4C000000 ff000000 10.12.3.4",`
	`cisco-avpair = "ipx:route=5C000000 ff000000 10.12.3.5"`

sap#	`cisco-avpair = "ipx:sap=4 CE1-LAB 1234.0000.0000.0001 451 4",` `cisco-avpair = "ipx:sap=5 CE3-LAB 2345.0000.0000.0001 452 5",`
sap-fltr-in#	`cisco-avpair = "ipx:sap-fltr-in=deny 6C01.0000.0000.0001",` `cisco-avpair = "ipx:sap-fltr-in=permit -1"`
sap-fltr-out#	`cisco-avpair = "ipx:sap-fltr-out=deny 6C01.0000.0000.0001",` `cisco-avpair = "ipx:sap-fltr-out=permit -1"`

pool-def#	`cisco-avpair = "ip:pool-def#1=aaa 10.0.0.1 1.0.0.3",`
	`cisco-avpair = "ip:pool-def#2=bbb 10.1.0.1 2.0.0.10",`
	`cisco-avpair = "ip:pool-def#3=ccc 10.2.0.1 3.0.0.20",`

pool-timeout	`cisco-avpair = "ip:pool-timeout=60"`

1. This attribute is specific to RADIUS servers. It can be used to add Cisco IOS interface configuration commands to specific user configuration information.

How to Configure a AAA Server for Per-User Configuration

The configuration requirements and the structure of per-user configuration information is set by the specifications of each type of AAA server. Refer to your server documentation for more detailed information. The following sections about TACACS and RADIUS servers are specific to per-user configuration:

- Configuring a Freeware TACACS Server for Per-User Configuration (As required)
- Configuring a CiscoSecure TACACS Server for Per-User Configuration (As required)
- Configuring a RADIUS Server for Per-User Configuration (As required)

See the section "Monitoring and Debugging Per-User Configuration Settings" later in this chapter for tips on troubleshooting per-user configuration settings. See the section "Configuration Examples for Per-User Configuration" at the end of this chapter for examples of configuring RADIUS and TACACS servers.

Configuring a Freeware TACACS Server for Per-User Configuration

On a TACACS server, the entry in the user file takes a standard form. In the freeware version of TACACS+, the following lines appear in order:

- "User =" followed by the username, a space, and an open brace
- Authentication parameters
- Authorization parameters
- One or more AV pairs
- End brace on a line by itself

The general form of a freeware TACACS user entry is shown in the following example:

```
user = username {
     authentication parameters go here
     authorization parameters go here
}
```

The freeware TACACS user entry form is also shown by the following examples for specific users:

```
user= Router1
  Password= cleartext welcome
  Service= PPP protocol= ip {
        ip:route=10.0.0.0 255.0.0.0
        ip:route=10.1.0.0 255.0.0.0
        ip:route=10.2.0.0 255.0.0.0
        ip:inacl#5=deny 10.5.0.1
}

user= Router2
  Password= cleartext lab
  Service= PPP protocol= ip {
        ip:addr-pool=bbb
}
```

For more requirements and detailed information, refer to your AAA server documentation.

Configuring a CiscoSecure TACACS Server for Per-User Configuration

The format of an entry in the user file in the AAA database is generally name = value. Some values allow additional subparameters to be specified and, in these cases, the subparameters are enclosed in braces ({}). The following simple example depicts an AAA database showing the default user, one group, two users that belong to the group, and one user that does not:

```
# Sample AA Database 1
unknown_user = {
  password = system #Use the system's password file (/etc/passwd)
}
group = staff {
  # Password for staff who do not have their own.
  password = des "sefjkAlM7zybE"
  service = shell {
    # Allow any commands with any attributes.
    default cmd = permit
    default attribute = permit
  }
}
user = joe {  # joe uses the group password.
```

```
  member = "staff"
}
user = pete {  # pete has his own password.
  member = "staff"
  password = des "alkd9Ujiqp2y"
}
user = anita {
  # Use the "default" user password mechanism defined above.
  service = shell {
    cmd = telnet {  # Allow Telnet to any destination
    }
  }
}
```

For more information about the requirements and details of configuring the CiscoSecure server, see the *CiscoSecure UNIX Server User Guide*.

Configuring a RADIUS Server for Per-User Configuration

On a RADIUS server, the format of an entry in the users file includes the following lines in order:

- Username and password
- User service type
- Framed protocol
- One or more AV pairs

Note All these AV pairs are vendor specific. To use them, RADIUS servers must support the use of vendor-specific AV pairs. Patches for some servers are available from the Cisco Consulting Engineering (CE) customer-support organization.

The structure of an AV pair for Cisco platforms starts with *cisco-avpair* followed by a space, an equal sign, and another space. The rest of the line is within double quotation marks and, for all lines but the last, ends with a comma. Inside the double quotation marks is a phrase indicating the supported attribute, another equal sign, and a Cisco IOS command. The following examples show two different partial user configurations on a RADIUS server.

Router1

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:route=10.0.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.1.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.2.0.0 255.0.0.0",
cisco-avpair = "ip:inacl#5=deny 10.5.0.1"
```

Router2

```
Password = "lab"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:addr-pool=bbb"
```

Monitoring and Debugging Per-User Configuration Settings

Per-user configuration information exists on AAA servers only and is configured there, as described in the "How to Configure a AAA Server for Per-User Configuration" section.

For more information about configuring an application that can tie AAA per-user configuration information to generic interface and router configuration, see the chapter "Configuring Virtual Profiles" in this publication. Virtual profiles are required for combining per-user configuration information and generic interface and router configuration information to create virtual access interfaces for individual ISDN B channels.

However, you can monitor and debug the per-user configuration settings on the router or access server that are set from an AAA server. Table 8 indicates some of the commands to use for each attribute.

Table 8 Monitoring and Debugging Per-User Configuration Commands

Attribute	show Commands	debug Commands
inacl# outacl#	**show ip access-list** **show ip interface** *interface* **show ipx access-list** **show ipx interface**	**debug aaa authorization** **debug aaa per-user**
rte-fltr-in# rte-fltr-out#	**show ip access-list** **show ip protocols**	**debug aaa authorization** **debug aaa per-user**
route#	**show ip route** **show ipx route**	**debug aaa authorization** **debug aaa per-user**
sap#	**show ipx servers**	**debug aaa authorization** **debug aaa per-user**
sap-fltr-in# sap-fltr-out#	**show ipx access-list** **show ipx interface**	**debug aaa authorization** **debug aaa per-user**
pool-def# pool-timeout	**show ip local pool** [*name*]	—

Configuration Examples for Per-User Configuration

The following sections provide two comprehensive examples:

- TACACS+ Freeware Examples
- RADIUS Examples

These examples show router or access server configuration and AV pair configuration on an AAA server.

TACACS+ Freeware Examples

This section provides the TACACS+ freeware versions of the following examples:

- IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI
- IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI

The following example provides configurations for the TACACS+ freeware daemon, the network access server, and the peer router named Router1. On the TACACS+ AAA server, peer router Router1 has a configuration that includes static routes and IP access lists.

TACACS+ Freeware Daemon Configuration File

```
key = tac123
user = Router1 {
global = cleartext welcome
service = ppp protocol = ip {
route#1=″10.0.0.0 255.0.0.0"
route#2=″10.1.0.0 255.0.0.0"
route#3=″10.2.0.0 255.0.0.0"
inacl#1=″deny 10.5.0.1"
}
}
```

Current Network Access Server Configuration

```
version 11.3
service timestamps debug datetime localtime
service udp-small-servers
service tcp-small-servers
!
hostname Router2
!
aaa new-model
aaa authentication ppp default tacacs+
aaa authorization network tacacs+
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
!
username Router1 password 7 15050E0007252621
ip host Router2 172.21.114.132
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
interface Ethernet 0
 ip address 172.21.114.132 255.255.255.224
 no ip mroute-cache
 media-type 10BaseT
!

interface Virtual-Template1
 ip unnumbered Ethernet0
 no cdp enable
!
!
interface BRI0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 no ip route-cache
 dialer idle-timeout 300
 dialer map ip 10.5.0.1 name Router1 broadcast 61482
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
!
```

```
ip default-gateway 172.21.114.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.114.129
!
virtual-profile virtual-template 1
dialer-list 1 protocol ip permit
tacacs-server host 172.21.114.130
tacacs-server key tac123
```

Current Peer Configuration for Router1

```
version 11.3
no service pad
!
hostname Router1
!
enable secret 5 $1$m1WK$RsjborN1Z.XZuFqsrtSnp/
enable password lab
!
username Router2 password 7 051C03032243430C
ip host Router1 172.21.114.134
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.21.114.134 255.255.255.224
 no ip route-cache
 shutdown
!
interface BRI0
 ip address 10.5.0.1 255.0.0.0
 encapsulation ppp
 dialer map ip 172.21.114.132 name Router2 broadcast 61483
 dialer-group 1
 no fair-queue
!
ip default-gateway 172.21.114.129
no ip classless
ip route 172.21.0.0 255.255.0.0 BRI0
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password lab
 login
end
```

IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

The following example provides configurations for the TACACS+ daemon and the peer router named Router1. On the TACACS+ AAA server, user ny has a configuration that includes inbound and outbound SAP filters.

TACACS+ Freeware Daemon Configuration File for User

```
key = tac123
user = Router1 {
 global = cleartext welcome
 service = ppp protocol = ipx {
        sap=”101 CYBER-01 40.0000.0000.0001 400 10"
        sap=”202 CYBER-02 40.0000.0000.0001 401 10"
        sap=”303 CYBER-03 40.0000.0000.0001 402 10"
        sap-fltr-out#1=”deny 40 101"
        sap-fltr-out#2=”deny 40 202"
        sap-fltr-out#3=”permit -1"
        sap-fltr-in#1=”permit 30 444"
        sap-fltr-in#2=”deny -1"
```

Current Remote Peer (Router1) Configuration

```
version 11.3
!
hostname Router1
!
enable password lab
!
username Router2 password 7 140017070F0B272E
ip host Router1 172.21.114.131
ip name-server 172.19.2.132
ip name-server 192.168.30.32
ipx routing 0000.0c47.090d
ipx internal-network 30
!
interface Ethernet0
 ip address 172.21.114.131 255.255.255.224
!
interface Serial1
 no ip address
 encapsulation ppp
 ipx ipxwan 0 unnumbered peer-Router1
 clockrate 4000000
!
ipx sap 444 ZEON-4 30.0000.0000.0001 444 10
ipx sap 555 ZEON-5 30.0000.0000.0001 555 10
ipx sap 666 ZEON-6 30.0000.0000.0001 666 10
!
Current Network Access Server (Router2) Configuration
version 11.3
service timestamps debug uptime
!
hostname Router2
!
aaa new-model
aaa authentication ppp default tacacs+
aaa authorization network tacacs+
enable password lab
!
username Router1 password 7 044C0E0A0C2E414B
ip host LA 172.21.114.133
ip name-server 192.168.30.32
```

```
ip name-server 172.19.2.132
ipx routing 0000.0c47.12d3
ipx internal-network 40
!
interface Ethernet0
 ip address 172.21.114.133 255.255.255.224
!
interface Virtual-Template1
 no ip address
 ipx ipxwan 0 unnumbered nas-Router2
 no cdp enable
!
interface Serial1
 ip unnumbered Ethernet0
 encapsulation ppp
 ipx ipxwan 0 unnumbered nas-Router2
 ppp authentication chap
!
ipx sap 333 DEEP9 40.0000.0000.0001 999 10
!
virtual-profile virtual-template 1
tacacs-server host 172.21.114.130
tacacs-server key tac123
```

RADIUS Examples

This section provides the RADIUS versions of the following examples:

- IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI
- IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

IP Access Lists and Static Routes Using Virtual Profiles over ISDN BRI

The following example shows a remote peer (Router1) configured to dial in to a BRI on a Cisco network access server (Router2), which requests user configuration information from an AAA server (radiusd):

RADIUS User File (Router1)

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ip:route=10.1.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.2.0.0 255.0.0.0",
cisco-avpair = "ip:route=10.3.0.0 255.0.0.0",
cisco-avpair = "ip:inacl#5=deny 10.0.0.1"
```

Current Network Access Server Configuration

```
version 11.3
service timestamps debug datetime localtime
service udp-small-servers
service tcp-small-servers
!
hostname Router2
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable secret 5 $1$koOn$/1QAylov6JFAElxRCrL.o/
enable password lab
```

```
!
username Router1 password 7 15050E0007252621
ip host Router2 172.21.114.132
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
interface Ethernet0
 ip address 172.21.114.132 255.255.255.224
 no ip mroute-cache
 media-type 10BaseT
!
interface Virtual-Template1
 ip unnumbered Ethernet0
 no cdp enable
!
interface BRI0
 ip unnumbered Ethernet0
 no ip mroute-cache
 encapsulation ppp
 no ip route-cache
 dialer idle-timeout 300
 dialer map ip 10.5.0.1 name Router1 broadcast 61482
 dialer-group 1
 no fair-queue
 ppp authentication chap
!
ip default-gateway 172.21.114.129
no ip classless
ip route 0.0.0.0 0.0.0.0 172.21.114.129
!
virtual-profile vtemplate 1
dialer-list 1 protocol ip permit
radius-server host 172.21.114.130
radius-server key rad123
```

Current Peer Configuration for Router1

```
version 11.3
no service pad
!
hostname Router1
!
enable secret 5 $1$m1WK$RsjborN1Z.XZuFqsrtSnp/
enable password lab
!
username Router2 password 7 051C03032243430C
ip host Router1 172.21.114.134
ip domain-name cisco.com
ip name-server 172.19.2.132
ip name-server 192.168.30.32
isdn switch-type basic-5ess
!
interface Ethernet0
 ip address 172.21.114.134 255.255.255.224
 no ip route-cache
 shutdown
!
interface BRI0
 ip address 10.5.0.1 255.0.0.0
 encapsulation ppp
 dialer map ip 172.21.114.132 name Router2 broadcast 61483
 dialer-group 1
 no fair-queue
```

```
!
ip default-gateway 172.21.114.129
no ip classless
ip route 172.21.0.0 255.255.0.0 BRI0
dialer-list 1 protocol ip permit
!
line con 0
 exec-timeout 0 0
line vty 0 4
 password lab
 login
!
end
```

Output of ping Command from Router1

```
Router1# ping 172.21.114.132

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.21.114.132, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

(fails due to access list deny)
```

RADIUS Debug Output

```
radrecv: Request from host ac157284 code=1, id=46, length=67
    Client-Id = 172.21.114.132
    Client-Port-Id = 1112670208
    User-Name = "Router1"
    CHAP-Password = "\037\317\213\326*\236)#+\266\243\255x\331\370v\334"
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
Sending Ack of id 46 to ac157284 (172.21.114.132)
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
    [Vendor 9] cisco-avpair = "ip:route=10.0.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:route=10.1.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:route=10.2.0.0 255.0.0.0"
    [Vendor 9] cisco-avpair = "ip:inacl#5=deny 10.0.0.1"
```

Network Access Server (Router2) show and debug Command Output

```
Router2# show debug

General OS:
  AAA Authorization debugging is on
PPP:
  PPP authentication debugging is on
  Multilink activity debugging is on
ISDN:
  ISDN events debugging is on
Dial on demand:
  Dial on demand events debugging is on
VTEMPLATE:
  Virtual Template debugging is on

pr  4 08:30:09: ISDN BR0: received HOST_INCOMING_CALL
        Bearer Capability i = 0x080010
*Apr  4 08:30:09:       -------------------
        Channel ID i = 0x0101
*Apr  4 08:30:09:         IE out of order or end of 'private' IEs --
        Bearer Capability i = 0x8890
```

```
*Apr  4 08:30:09:         Channel ID i = 0x89
*Apr  4 08:30:09:         Called Party Number i = 0xC1, '61483'
*Apr  4 08:30:09: ISDN BR0: Event: Received a call from <unknown> on B1 at 64 Kb/s
*Apr  4 08:30:09: ISDN BR0: Event: Accepting the call
%LINK-3-UPDOWN: Interface BRI0:1, changed state to up
*Apr  4 08:30:09: ISDN BR0: received HOST_CONNECT
        Channel ID i = 0x0101
*Apr  4 08:30:09:       -------------------
        Channel ID i = 0x89
*Apr  4 08:30:09: ISDN BR0: Event: Connected to <unknown> on B1 at 64 Kb/s
*Apr  4 08:30:09: PPP BRI0:1: Send CHAP challenge id=30 to remote
*Apr  4 08:30:10: PPP BRI0:1: CHAP response received from Router1
*Apr  4 08:30:10: PPP BRI0:1: CHAP response id=30 received from Router1
*Apr  4 08:30:10: AAA/AUTHOR/LCP: authorize LCP
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (0): send AV protocol=lcp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: BRI0:1: (2084553184): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (2084553184): Post authorization status = PASS_ADD
*Apr  4 08:30:10: PPP BRI0:1: Send CHAP success id=30 to remote
*Apr  4 08:30:10: PPP BRI0:1: remote passed CHAP authentication.
*Apr  4 08:30:10: VTEMPLATE Reuse vaccess1, New Recycle queue size:0

*Apr  4 08:30:10: VTEMPLATE set default vaccess1 with no ip address

*Apr  4 08:30:10: Virtual-Access1 VTEMPLATE hardware address 0000.0c46.154a
*Apr  4 08:30:10: VTEMPLATE vaccess1 has a new cloneblk vtemplate, now it has vtemplate
*Apr  4 08:30:10: VTEMPLATE undo default settings vaccess1

*Apr  4 08:30:10: VTEMPLATE ************* CLONE VACCESS1 ******************Apr  4
08:30:10: VTEMPLATE Clone from vtemplate1 to vaccess1
interface Virtual-Access1
no ip address
encap ppp
ip unnumbered ethernet 0
end

%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Apr  4 08:30:10: AAA/AUTHOR/LCP: authorize LCP
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (0): send AV protocol=lcp
*Apr  4 08:30:10: AAA/AUTHOR/LCP: Virtual-Access1: (1338953760): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (1338953760): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): can we start IPCP?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV protocol=ip
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (1716082074): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (1716082074): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: we can start IPCP (0x8021)
*Apr  4 08:30:10: MLP Bad link Virtual-Access1
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): can we start UNKNOWN?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (0): send AV protocol=unknown
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: (2526612868): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (2526612868): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: Virtual-Access1: we can start UNKNOWN (0x8207)
*Apr  4 08:30:10: MLP Bad link Virtual-Access1
*Apr  4 08:30:10: BRI0:1: Vaccess started from dialer_remote_name
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): can we start IPCP?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV service=ppp
```

```
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV protocol=ip
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (3920403585): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (3920403585): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: we can start IPCP (0x8021)
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): can we start UNKNOWN?
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): user='Router1'
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV service=ppp
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (0): send AV protocol=unknown
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: (3439943223): Method=RADIUS
*Apr  4 08:30:10: AAA/AUTHOR (3439943223): Post authorization status = PASS_ADD
*Apr  4 08:30:10: AAA/AUTHOR/FSM: BRI0:1: we can start UNKNOWN (0x8207)
%LINEPROTO-5-UPDOWN: Line protocol on Interface BRI0:1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: start: her address 10.0.0.1, we want
0.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): user='Router1'
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV servi*Apr  4 08:30:13:
AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV protocol=ip
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (0): send AV addr*10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: (3215797579): Method=RADIUS
*Apr  4 08:30:13: AAA/AUTHOR (3215797579): Post authorization status = PASS_ADD
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV protocol=ip
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV addr*10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.1.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.2.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV route=10.3.0.0 255.0.0.0
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: Processing AV inacl#5=deny 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: authorization succeeded
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: done: her address 10.0.0.1, we want
10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR/IPCP: Virtual-Access1: authorization succeeded
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 10.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 10.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 11.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 11.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: parse_cmd 'ip route 12.0.0.0 255.0.0.0
10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip route 12.0.0.0
255.0.0.0 10.0.0.1
*Apr  4 08:30:13: AAA/AUTHOR: parse 'ip access-list standard Virtual-Access1#1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: parse 'deny 10.0.0.1' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: enqueue peruser IP txt=no ip access-list
standard Virtual-Access1#1
*Apr  4 08:30:13: VTEMPLATE vaccess1 has a new cloneblk AAA, now it has vtemplate/AAA
*Apr  4 08:30:13: VTEMPLATE ************* CLONE VACCESS1 *****************

*Apr  4 08:30:13: VTEMPLATE Clone from AAA to vaccess1
interface Virtual-Access1
ip access-group Virtual-Access1#1 in

*Apr  4 08:30:13: AAA/AUTHOR: Virtual-Access1: vaccess parse 'interface Virtual-Access1
ip access-group Virtual-Access1#1 in
' ok (0)
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Check for unauthorized mandatory AV's
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Processing AV service=ppp
*Apr  4 08:30:13: AAA/AUTHOR/FSM: Processing AV protocol=unknown
*Apr  4 08:30:13: AAA/AUTHOR/FSM: succeeded
%ISDN-6-CONNECT: Interface BRI0:1 is now connected to Router1
```

```
Router2# show ip access-list

Standard IP access list Virtual-Access1#1 (per-user)
    deny   10.0.0.1

Router2# show ip route

Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.21.114.129 to network 0.0.0.0

U    10.0.0.0/8 [1/0] via 10.3.0.1
U    10.1.0.0/8 [1/0] via 10.3.0.1
U    10.2.0.0/8 [1/0] via 10.3.0.1
     10.3.0.0/8 is subnetted, 1 subnets
C       10.3.0.1 is directly connected, Virtual-Access1
     172.21.0.0/16 is subnetted, 1 subnets
C       172.21.114.128 is directly connected, Ethernet0
S*   0.0.0.0/0 [1/0] via 172.21.114.129


Router2# show interfaces virtual-access 1

Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered.  Using address of Ethernet0 (172.21.114.132)
  MTU 1500 bytes, BW 64 Kbit, DLY 100000 usec, rely 255/255, load 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open, multilink Closed
  Open: IPCP, CDP
  Last input 5d04h, output never, output hang never
  Last clearing of "show interface" counters 00:06:42
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     76 packets input, 3658 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     141 packets output, 2909 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions

Router2# show ip interface virtual-access 1

Virtual-Access1 is up, line protocol is up
  Interface is unnumbered.  Using address of Ethernet0 (172.21.114.132)
  Broadcast address is 255.255.255.255
  Peer address is 10.0.0.1
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is enabled
  Outgoing access list is not set
  Inbound  access list is Virtual-Access1#1
  Proxy ARP is enabled
  Security level is default
```

```
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
  IP fast switching is disabled


Router2# debug ip packet

IP packet debugging is on
Router2#
*Apr  4 08:30:42: IP: s=172.21.114.129 (Ethernet0), d=255.255.255.255, len 186, rcvd 2
*Apr  4 08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, a*Apr  4
08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access denied
*Apr  4 08:30:42: IP: s=172.21.114.132 (local), d=10.0.0.1 (Virtual-Access1), len 4,
sending
*Apr  4 08:30:42: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
*Apr  4 08:30:44: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
*Apr  4 08:30:44: IP: s=172.21.114.132 (local), d=10.0.0.1 (Virtual-Access1), len 16,
sending
*Apr  4 08:30:44: IP: s=10.0.0.1 (Virtual-Access1), d=172.21.114.132, len 104, access
denied
```

IPX Per-User SAP Filters Using IPXWAN and Virtual Profiles by a Synchronous Interface

The following examples show a remote peer (Router1) configured to dial in to a synchronous interface on a Cisco network access server (Router2), which requests user configuration information from an AAA server (radiusd):

RADIUS User File (Router 1)

```
Password = "welcome"
User-Service-Type = Framed-User,
Framed-Protocol = PPP,
cisco-avpair = "ipx:sap=101 CYBER-01 40.0000.0000.0001 400 10",
cisco-avpair = "ipx:sap=202 CYBER-02 40.0000.0000.0001 401 10",
cisco-avpair = "ipx:sap=303 CYBER-03 40.0000.0000.0001 402 10",
cisco-avpair = "ipx:sap-fltr-out#20=deny 40 101",
cisco-avpair = "ipx:sap-fltr-out#21=deny 40 202",
cisco-avpair = "ipx:sap-fltr-out#22=permit -1",
cisco-avpair = "ipx:sap-fltr-in#23=permit 30 444",
cisco-avpair = "ipx:sap-fltr-in#23=deny -1"
```

Current Remote Peer (Router 1) Configuration

```
hostname Router1
!
enable password lab
!
username Router2 password 7 140017070F0B272E
ip host Router1 172.21.114.131
ip name-server 172.19.2.132
ip name-server 192.168.30.32
ipx routing 0000.0c47.090d
ipx internal-network 30
!
interface Ethernet0
 ip address 172.21.114.131 255.255.255.224
!
interface Serial1
```

```
 no ip address
 encapsulation ppp
 ipx ipxwan 0 unnumbered peer-Router1
 clockrate 4000000
!
ipx sap 444 ZEON-4 30.0000.0000.0001 444 10
ipx sap 555 ZEON-5 30.0000.0000.0001 555 10
ipx sap 666 ZEON-6 30.0000.0000.0001 666 10
!
...
version 12.1
service timestamps debug uptime
!
hostname Router2
!
aaa new-model
aaa authentication ppp default radius
aaa authorization network radius
enable password lab
!
username Router1 password 7 044C0E0A0C2E414B
ip host Router2 172.21.114.133
ip name-server 172.22.30.32
ip name-server 192.168.2.132
ipx routing 0000.0c47.12d3
ipx internal-network 40
!
interface Ethernet0
 ip address 172.21.114.133 255.255.255.224
!
interface Virtual-Template1
 no ip address
 ipx ipxwan 0 unnumbered nas-Router2
 no cdp enable
!
interface Serial1
 ip unnumbered Ethernet0
 encapsulation ppp
 ipx ipxwan 0 unnumbered nas-Router2
 ppp authentication chap
!
ipx sap 333 DEEP9 40.0000.0000.0001 999 10
!
virtual-profile vtemplate 1
radius-server host 172.21.114.130
radius-server key rad123
```

RADIUS debug Output

```
radrecv: Request from host ac157285 code=1, id=23, length=67
    Client-Id = 172.21.114.133
    Client-Port-Id = 1399128065
    User-Name = "Router1"
    CHAP-Password = "%"(\012I$\262\352\031\276\024\302\277\225\347z\274"
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
Sending Ack of id 23 to ac157285 (172.21.114.133)
    User-Service-Type = Framed-User
    Framed-Protocol = PPP
    [Vendor 9] cisco-avpair = "ipx:sap=101 CYBER-01 40.0000.0000.0001 400 10"
    [Vendor 9] cisco-avpair = "ipx:sap=202 CYBER-02 40.0000.0000.0001 401 10"
    [Vendor 9] cisco-avpair = "ipx:sap=303 CYBER-03 40.0000.0000.0001 402 10"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#20=deny1 40 101"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#21=deny 40 202"
```

```
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-out#22=permit -1"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-in#23=permit 30 444"
    [Vendor 9] cisco-avpair = "ipx:sap-fltr-in#23=deny -1"
```

Network Access Server show Command Output

```
Router2# show ipx servers

Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail
5 Total IPX Servers

Table ordering is based on routing and server info

   Type Name                          Net     Address    Port     Route Hops Itf
s   101 CYBER-01                      40.0000.0000.0001:0400      conn  10  Int
s   202 CYBER-02                      40.0000.0000.0001:0401      conn  10  Int
s   303 CYBER-03                      40.0000.0000.0001:0402      conn  10  Int
S   333 DEEP9                         40.0000.0000.0001:0999      conn  10  Int
P   444 ZEON-4                        30.0000.0000.0001:0444      7/01  11  Vi1


Router1# show ipx servers

Codes: S - Static, P - Periodic, E - EIGRP, N - NLSP, H - Holddown, + = detail
5 Total IPX Servers

Table ordering is based on routing and server info

   Type Name                          Net     Address    Port     Route Hops Itf
P   303 CYBER-03                      40.0000.0000.0001:0402      7/01  11  Se1
P   333 DEEP9                         40.0000.0000.0001:0999      7/01  11  Se1
S   444 ZEON-4                        30.0000.0000.0001:0444      conn  10  Int
S   555 ZEON-5                        30.0000.0000.0001:0555      conn  10  Int
S   666 ZEON-6                        30.0000.0000.0001:0666      conn  10  Int

Router2# show ipx access-list

IPX sap access list Virtual-Access1#2
```

Configuring Resource Pool Management

This chapter describes the Cisco Resource Pool Management (RPM) feature. It includes the following main sections:

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

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

RPM Overview

Cisco RPM enables telephone companies and Internet service providers (ISPs) to share dial resources for wholesale and retail dial network services. With RPM, telcos and ISPs can count, control, and manage dial resources and provide accounting for shared resources when implementing different service-level agreements.

You can configure RPM in a single, standalone Cisco network access server (NAS) by using RPM or, optionally, across multiple NAS stacks by using one or more external Cisco Resource Pool Manager Servers (RPMS).

Cisco RPM gives data network service providers the capability to do the following:

- Have the flexibility to include local retail dial services in the same NAS with the wholesale dial customers.
- Manage customer use of shared resources such as modems or High-Level Data Link Control (HDLC) controllers for data calls.

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

- Offer advanced wholesale dialup services using a Virtual Private Dialup Network (VPDN) to enterprise accounts and ISPs.
- Deploy Data over Voice Bearer Service (DoVBS).
- Manage call sessions by differentiating dial customers through customer profiles. The customer profile determines where resources are allocated and is based on the incoming Dialed Number Information Service (DNIS) number or Calling Line Identification (CLID).
- Efficiently use resource groups such as modems to offer differing over subscription rates and dial service-level agreements.

Note Ear and Mouth Feature Group B (E&M-FGB) is the only signaling type supported for channel-associated signaling (CAS) on T1 and T3 facilities; R2 is supported for E1 facilities. FG D is not supported. Cisco IOS software collects DNIS digits for the signaling types FGB, PRI, and SS7 and only E&M-FGB and R2 CAS customer profiles are supported. For all other CAS signaling types, use the default DNIS group customer profiles.

Components of Incoming and Outgoing Call Management

Cisco RPM manages both incoming calls and outgoing sessions. Cisco RPM differentiates dial customers through configured customer profiles based on the DNIS and call type determined at the time of an incoming call.

The components of incoming call management in the Cisco RPM are described in the following sections:

- Customer Profile Types
- DNIS Groups
- Call Types
- Resource Groups
- Resource Services

You can use Cisco RPM to answer all calls and differentiate customers by using VPDN profiles and groups. The components of outgoing session management in the Cisco RPM are described in the following sections:

- VPDN Groups
- VPDN Profiles

Note These components of Cisco RPM are enabled after the NAS and other equipment has been initially set up, configured, and verified for proper operation of the dial, PPP, VPDN, and authentication, authorization, and accounting (AAA) segments. Refer to the Cisco IOS documentation for these other segments for installation, configuration, and troubleshooting information before attempting to use RPM.

Configured DNIS groups and resource data can be associated to customer profiles. These customer profiles are selected by the incoming call DNIS number and call type and then used to identify resource allocations based on the associated resource groups and defined resource services.

After the call is answered, customer profiles can also be associated with VPDN groups so the configured VPDN sessions and other data necessary to set up or reject a VPDN session are applied to the answered calls. VPDN group data includes associated domain name or DNIS, IP addresses of endpoints, maximum sessions per endpoint, maximum Multilink PPP (MLP) bundles per VPDN group, maximum links per MLP bundle, and other tunnel information.

Customer Profile Types

There are three types of customer profiles in Cisco RPM, which are described in the following sections:

- Customer Profiles
- Default Customer Profiles
- Backup Customer Profiles

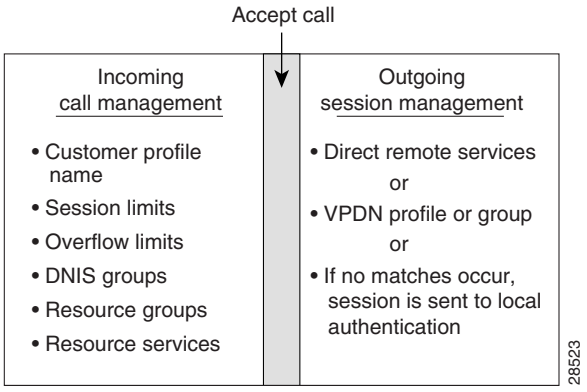
Additionally, you can create a customer profile template and associate it with a customer profile; it is then integrated into the customer profile.

Customer Profiles

A customer profile defines how and when to answer a call. Customer profiles include the following components (see Figure 35):

- Customer profile name and description—Name and description of the customer.
- Session limits—Maximum number of standard sessions.
- Overflow limits—Maximum number of overflow sessions.
- DNIS groups.
- CLID.
- Resource groups.
- Resource services.
- VPDN groups and VPDN profiles.
- Call treatment—Determines how calls that exceed the session and overflow limits are treated.

Figure 35 Components of a Customer Profile



The incoming side of the customer profile determines if the call will be answered using parameters such as DNIS and call type from the assigned DNIS group and session limits. The call is then assigned the appropriate resource within the resource group defined in the customer profile. Each configured customer profile includes a maximum allowed session value and an overflow value. As sessions are started and ended, session counters are incremented and decremented so customer status is kept current. This information is used to monitor the customer resource limit and determine the appropriate call treatment based on the configured session limits.

The outgoing side of the customer profile directs the answered call to the appropriate destination:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

- To a local AAA server of retail dial applications and Internet/intranet access.
- To a tunnel that is established between the NAS or L2TP Access Concentrator (LAC) to a wholesale VPDN home gateway of a dial customer, or L2TP Network Server (LNS) using Layer 2 Forwarding Protocol (L2F) or Layer 2 Tunneling Protocol (L2TP) technology.

Default Customer Profiles

Default customer profiles are identical to standard customer profiles, except that they do not have any associated DNIS groups. Default customer profiles are created using the reserved keyword **default** for the DNIS group.

Default customer profiles are used to provide session counting and resource assignment to incoming calls that do not match any of the configured DNIS groups. Although specific resources and DNIS groups can be assigned to customer profiles, default customer profiles allow resource pooling for the calls that do not match the configured DNIS groups or where the DNIS is not provided. Retail dial services and domain-based VPDN use default customer profiles.

When multiple default customer profiles are used, the call type (speech, digital, V.110, or V.120) of the default DNIS group is used to identify which default customer profile to use for an incoming call. At most, four default profiles (one for each call type) can be configured.

Note If default customer profiles are not defined, then calls that do not match a DNIS group in a customer profile are rejected with a “no answer” or “busy” call treatment sent to the switch.

Backup Customer Profiles

Backup customer profiles are customer profiles configured locally on the Cisco NAS and are used to answer calls based on a configured allocation scheme when the link between the Cisco NAS and Cisco RPMS is disabled. See the section “Configuring Customer Profiles Using Backup Customer Profiles” for more information about configuring backup customer profiles.

Customer Profile Template

With RPM, users can also implement wholesale dial services without using VPDN tunnels to complete dial-in calls to destinations of the end customer. This capability is accomplished with components of the AAA groups and the PPP configurations.

The AAA group provides IP addresses of AAA servers for authentication and accounting. The PPP configurations allow users to configure the Cisco IOS PPP feature set on each customer profile. In this current implementation, PPP configuration is based on the following:

- Applicable IP address pool(s) or default local list of IP addresses
- Primary and secondary Domain Name System (DNS) or Windows Internet naming service (WINS)
- Number of links allowed for each call using MLP

Note The AAA and PPP integration applies to a single NAS environment.

To add PPP configurations to a customer profile, you must create a customer profile template. Once you create the template and associate it with a customer profile using the **source template** command, it is integrated into the customer profile.

The RPM customer profile template for the PPP command set, when used with the Cisco IOS feature, Server Groups Selected by DNIS, presents a strong single NAS solution for providers of wholesale dial services, as follows:

- Call acceptance is determined by the RPM before call answering, using the configured size limits and resource availability.
- The answered call then uses the PPP configuration defined in the template to initiate authentication, obtain an IP address, and select a DNS or WINS that is located at the customer site.
- The same DNIS that was used to choose the customer profile selects the servers for authentication/authorization and accounting that are located at the wholesale customer's site.

The section "Configuring a Customer Profile Template" later in this chapter describes how to create a customer profile template so that you can configure the Cisco IOS PPP features on a customer profile, but this section does not list the existing PPP command set. For information about the PPP command set, refer to the *Cisco IOS Dial Technologies Command Reference*.

DNIS Groups

A DNIS group is a configured list of DNIS called party numbers that correspond to the numbers dialed to access particular customers, service offerings, or both. For example, if a customer from phone number 000-1234 calls a number 000-5678, the DNIS provides information on the number dialed—000-5678.

Cisco RPM checks the DNIS number of inbound calls against the configured DNIS groups, as follows:

- If Cisco RPM finds a match, it uses the configured information in the customer profile to which the DNIS group is assigned.
- If Cisco RPM does not find a match, it uses the configured information in the customer profile to which the default DNIS group is assigned.
- The DNIS/call type sequence can be associated only with one customer profile.

CLID Groups

A CLID group is a configured list of CLID calling party numbers. The CLID group specifies a list of numbers to reject if the group is associated with a call discriminator. For example, if a customer from phone number 000-1234 calls a number 000-5678, the CLID provides information on the calling party number—000-1234.

A CLID can be associated with only one CLID group.

Call Types

Call types from calls originating from ISDN, SS7, and CAS (CT1, CT3, and CE1) are used to assign calls to the appropriate resource. Call types for ISDN and SS7 are based on Q.931 bearer capability. Call types for CAS are assigned based on static channel configuration.

Supported call types are as follows:

- Speech
- Digital
- V.110
- V.120

Note Voice over IP, fax over IP, and dial-out calls are not supported in RPM.

Resource Groups

Cisco RPM enables you to maximize the use of available shared resources within a Cisco NAS for various resource allocation schemes to support service-level agreements. Cisco RPM allows you to combine your Cisco NAS resource groups with call types (speech, digital, V.110, and V.120) and optional resource modem services. Resource groups and services are configured for customer profiles and assigned to incoming calls through DNIS groups and call types.

Resource groups have the following characteristics:

- Are configured on the Cisco NAS and applied to a customer profile.
- Represent groupings of similar hardware or firmware that are static and do not change on a per-call basis.
- Can define resources that are port-based or not port-based:
  - Port-based resources are identified by physical location, such as a range of port/slot numbers (for example, modems or terminal adapters).
  - Non-port-based resources are identified by a single size parameter (for example, HDLC framers or V.120 terminal adapters—V.120 terminal adapters are currently implemented as part of Cisco IOS software).

Resource assignments contain combinations of Cisco NAS resource groups, optional resource modem services, and call types. The NAS resources in resource groups that have not been assigned to a customer profile will not be used.

Note To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource. The resource group assigned to this customer profile will be “digital resources” and also have a call type of “speech,” so the call will terminate on an HDLC controller rather than a modem.

Resource Services

A resource service contains a finite series of resource command strings that can be used to help dynamically configure an incoming connection. Services supported by a resource group are determined by the combination of hardware and firmware installed. Currently, resource service options can be configured and applied to resource groups. Resource services can be defined to affect minimum and maximum speed, modulation, error correction, and compression, as shown in Table 9.

Table 9 Resource Services

Service	Options	Comments
min-speed	<300–56000>, any	Must be a V.90 increment.
max-speed	<300–56000>, any	Must be a V.90 increment.
modulation	k56flex, v22bis, v32bis, v34, v90, any	None.
error-correction	1apm, mn14	This is a hidden command.
compression	mnps, v42bis	This is a hidden command.

VPDN Groups

The VPDN group contains the data required to build a VPDN tunnel from the RPM NAS LAC to the LNS. In the context of RPM, VPDN is authorized by first associating a customer profile with a VPDN group, and second by associating the VPDN group to the DNIS group used for that customer profile. VPDN group data includes the endpoint IP addresses.

Cisco RPM enables you to specify multiple IP endpoints for a VPDN group, as follows:

- If two or more IP endpoints are specified, Cisco RPM uses a load-balancing method to ensure that traffic is distributed across the IP endpoints.
- For DNIS-based VPDN dial service, VPDN groups are assigned to customer profiles based on the incoming DNIS number and the configured DNIS groups.
- For domain-based VPDN dial service, VPDN groups are assigned to the customer profile or the default customer profile with the matching call-type assignment.
- For either DNIS-based or domain-based VPDN dial services, there is a customer profile or default customer profile for the initial resource allocation and customer session limits.

The VPDN group provides call management by allowing limits to be applied to both the number of MLP bundles per tunnel and the number of links per MLP bundle. Limits can also restrict the number of sessions per IP endpoint. If you require more granular control of VPDN counters, use VPDN profiles.

VPDN Profiles

VPDN profiles allow session and overflow limits to be imposed for a particular customer profile. These limits are unrelated to the limits imposed by the customer profile. A customer profile is associated with a VPDN profile. A VPDN profile is associated with a VPDN group. VPDN profiles are required only when these additional counters are required for VPDN usage per customer profile.

Call Treatments

Call treatment determines how calls are handled when certain events require the call to be rejected. For example, if the session and overflow limits for one of your customers have been exceeded, any additional calls will receive a busy signal (see Table 10).

Table 10 Call-Treatment Table

Event	Call-Treatment Option	Results
Customer profile not found	No answer (default)	The caller receives rings until the switch eventually times out. Implies that the NAS was appropriate, but resources were unavailable. The caller should try later.
	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller. The call is rejected based on not matching a DNIS group/call type and customer profile. Can be used to immediately reject the call and free up the circuit.
Customer profile limits exceeded	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller.
NAS resource not available	Channel not available (default)	The switch sends the call to the next channel in the trunk group. The call can be answered, but the NAS does not have any available resources in the resource groups. Allows the switch to try additional channels until it gets to a different NAS in the same trunk group that has the available resources.
	Busy	The switch drops the call from the NAS and sends a busy signal back to the caller. Can be used when the trunk group does not span additional NASes.
Call discrimination match	No answer	The caller receives rings until the switch eventually times out.

Details on RPM Call Processes

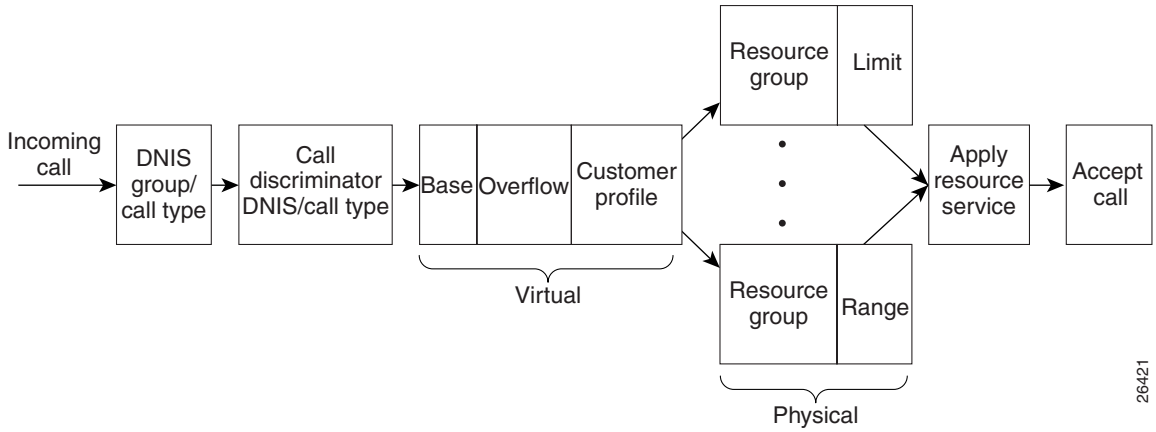
On the incoming call management of the customer profile, the following sequence occurs to determine if a call is answered:

1. The incoming DNIS is mapped to a DNIS group; if there is no incoming DNIS number, or the DNIS number provided does not match any configured DNIS group, the DNIS group *default* is used.
2. The mapped DNIS group is checked against configured call discriminator profiles to confirm if this DNIS group/call-type combination is disallowed. If there is a match, the call is immediately rejected.
3. Once a DNIS group or a default DNIS group is identified, the customer profile associated with that DNIS group and the call type (from the bearer capability for ISDN call, statically configured for CAS calls) is selected. If there is no corresponding customer profile, the call is rejected.

4. The customer profile includes a session limit value and an overflow limit value. If these thresholds are not met, the call is then assigned the appropriate resource defined in the customer profile. If the thresholds are met, the call is rejected.
5. If resources are available from the resource group defined in the customer profile, the call is answered. Otherwise, the call is rejected.
6. As sessions start and end, the session counters increase and decrease, so the customer profile call counters are kept current.

See Figure 36 for a graphical illustration of the RPM call processes.

Figure 36 Incoming Call Management: RPM Functional Description

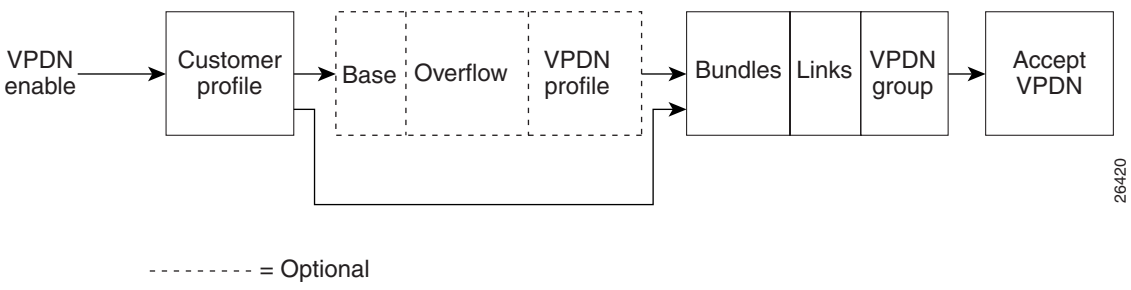


After the call is answered and if VPDN is enabled, Cisco RPM checks the customer profile for an assigned VPDN group or profile. The outgoing session management of the customer profile directs the answered call to the appropriate destination (see Figure 37), as follows:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

- To a local AAA server of retail dial applications and Internet/intranet access.
- To a tunnel that is established between the NAS or LAC and a wholesale VPDN home gateway from a dial customer or LNS using L2F or L2TP tunneling technology.

Figure 37 Outgoing Call Management: RPM Functional Description for VPDN Profiles and Groups



If a VPDN profile is found, the limits are checked, as follows:

- If the limits have not been exceeded, the VPDN group data associated with that VPDN profile is used to build a VPDN tunnel.
- If the VPDN limits have been exceeded, the call is disconnected.

If a VPDN group is found within the customer profile, the VPDN group data is used to build a VPDN tunnel, as follows:

- If the VPDN group limits (number of multilink bundles, number of links per bundle) have not been exceeded, a VPDN tunnel is built.
- If the limits have been reached, the call is disconnected.

If no VPDN profile is assigned to the customer profile and VPDN is enabled, non-RPM VPDN service is attempted. If the attempt fails, the call is processed as a retail dial service call if local AAA service is available.

Accounting Data

You can generate accounting data for network dial service usage in NAS AAA attribute format.

You can configure the Cisco NAS to generate AAA accounting records for access to external AAA server option. The accounting start and stop records in AAA attribute format are sent to the external AAA server using either RADIUS server hosts or TACACS+ protocols for accounting data storage. Table 11 lists the new fields in the AAA accounting packets.

Table 11 AAA Accounting Records

Accounting Start Record	Accounting Stop Record
Call-Type	Disconnect-Cause
CAS-Group-Name	Modem-Speed-Receive
Customer-Profile-Name	Modem-Speed-Transmit
Customer-Profile-Active-Sessions	MLP-Session-ID
DNIS-Group-Name	
Overflow	
MLP-Session_ID	
Modem-Speed-Receive	
Modem-Speed-Transmit	
VPDN-Domain-Name	
VPDN-Tunnel-ID	
VPDN-HomeGateway	
VPDN-Group-Active-Sessions	

Data over Voice Bearer Services

DoVBS is a dial service that uses a customer profile and an associated resource group of digital resources to direct data calls with a speech call type to HDLC controllers.

To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource.

The resource group assigned to this customer profile will be “digital resources” and will also have a call type of speech, so the call will terminate on an HDLC controller rather than a modem.

Call Discriminator Profiles

The Cisco RPM CLID/DNIS Call Discriminator feature lets you specify a list of calling party numbers to be rejected for inbound calls. This Cisco IOS Release 12.2 CLID/DNIS call screening feature expands previous call screening features in Cisco RPM. CLID/DNIS call screening provides an additional way to screen calls on the basis of CLID/DNIS for both local and remote RPM.

Cisco RPM CLID/DNIS Call Discriminator profiles enable you to process calls differently on the basis of the call type and CLID combination. Resource pool management offers a call discrimination feature that rejects calls on the basis of a CLID group and a call type filter. When a call arrives at the NAS, the CLID and the call type are matched against a table of disallowed calls. If the CLID and call type match entries in this table, the call is rejected before it is assigned Cisco NAS resources or before any other Cisco RPM processing occurs. This is called precall screening.

Precall screening decides whether the call is allowed to be processed. You can use the following types of discriminators to execute precall screening:

- ISDN discriminator—Accepts a call if the calling number matches a number in a group of configured numbers (ISDN group). This is also called white box screening. If you configure an ISDN group, only the calling numbers specified in the group are accepted.
- DNIS discriminator—Accepts a call if the called party number matches a number in a group of configured numbers (DNIS group). If you set up a DNIS group, only the called party numbers in the group are accepted. DNIS gives you information about the called party.
- Cisco RPM CLID/DNIS discriminator—Rejects a call if the calling number matches a number in a group of configured numbers (CLID/DNIS group). This is also called black box screening.

  If you configure a discriminator with a CLID group, the calling party numbers specified in the group are rejected. CLID gives you information about the caller.

  Similarly, if you configure a discriminator with a DNIS group, the called party numbers specified in the group are rejected.

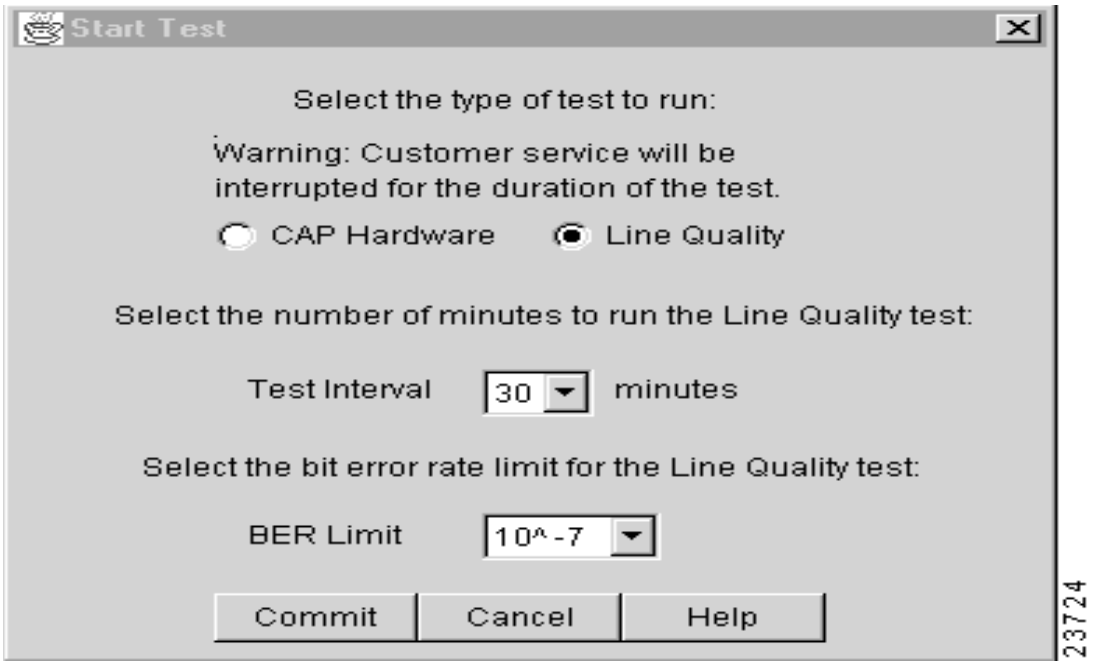
The Cisco RPM CLID/DNIS Call Discriminator Feature is independent of ISDN or DNIS screening done by other subsystems. ISDN or DNIS screening and Cisco RPM CLID/DNIS screening can both be present in the same system. Both features are executed if configured. Similarly, if DNIS Preauthorization using AAA is configured, it is present in addition to Cisco RPM CLID/DNIS screening. Refer to the *Cisco IOS Security Configuration Guide* for more information about call preauthorization.

In Cisco RPM CLID/DNIS screening, the discriminator can be a CLID discriminator, a DNIS discriminator, or a discriminator that screens on both the CLID and DNIS. The resulting discrimination logic is:

- If a discriminator contains just DNIS groups, it is a DNIS discriminator that ignores CLID. The DNIS discriminator blocks the call if the called number is in a DNIS group, which the call type references.
- If a discriminator contains just CLID groups, it is a CLID discriminator that ignores DNIS. The CLID discriminator blocks the call if the calling number is in a CLID group, which the call type references.
- If a discriminator contains both CLID and DNIS groups, it is a logical AND discriminator. It blocks the call if the calling number and called number are in the CLID or DNIS group, and the call type references the corresponding discriminator.

Figure 38 shows how call discrimination can be used to restrict a specific DNIS group to only modem calls by creating call discrimination settings for the DNIS group and the other supported call types (digital, V.110, and V.120).

Figure 38 Call Discrimination



Incoming Call Preauthentication

With ISDN PRI or channel-associated signaling (CAS), information about an incoming call is available to the NAS before the call is connected. The available call information includes:

- The DNIS, also referred to as the *called number*
- The CLID, also referred to as the *calling number*
- The call type, also referred to as the *bearer capability*

The Preauthentication with ISDN PRI and Channel-Associated Signalling feature introduced in Cisco IOS Release 12.2 allows a Cisco NAS to decide—on the basis of the DNIS number, the CLID number, or the call type—whether to connect an incoming call.

When an incoming call arrives from the public network switch, but before it is connected, this feature enables the NAS to send the DNIS number, CLID number, and call type to a RADIUS server for authorization. If the server authorizes the call, the NAS accepts the call. If the server does not authorize the call, the NAS sends a disconnect message to the public network switch to reject the call.

The Preauthentication with ISDN PRI and Channel-Associated Signalling feature offers the following benefits:

- With ISDN PRI, it enables user authentication and authorization before a call is answered. With CAS, the call must be answered; however, the call can be dropped if preauthentication fails.
- It enables service providers to better manage ports using their existing RADIUS solutions.
- Coupled with a preauthentication RADIUS server application, it enables service providers to efficiently manage the use of shared resources to offer differing service-level agreements.

For more information about the Preauthentication with ISDN PRI and Channel-Associated Signalling feature, refer to the *Cisco IOS Security Configuration Guide*.

RPM Standalone Network Access Server

A single NAS using Cisco RPM can provide the following:

- Wholesale VPDN dial service to corporate customers
- Direct remote services
- Retail dial service to end users

Figure 39 and Figure 40 show multiple connections to a Cisco AS5300 NAS. Incoming calls to the NAS can use ISDN PRI signaling, CAS, or the SS7 signaling protocol. Figure 39 shows incoming calls that are authenticated locally for retail dial services or forwarded through VPDN tunnels for wholesale dial services.

Note This implementation does not use Cisco RPM CLID/DNIS Call Discriminator Feature. If you are not using Cisco RPMS and you have more than one Cisco NAS, you must manually configure each NAS by using Cisco IOS commands. Resource usage information is not shared between NASes.

Figure 39 Retail Dial Service Using RPM

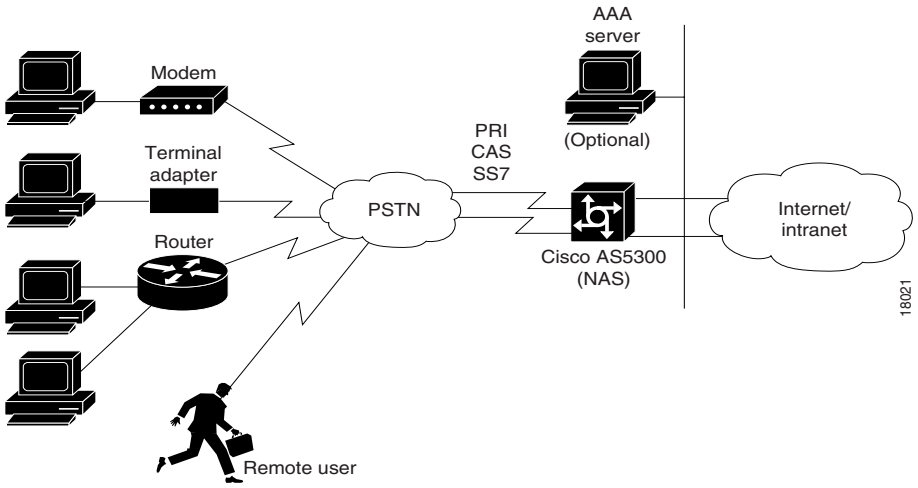
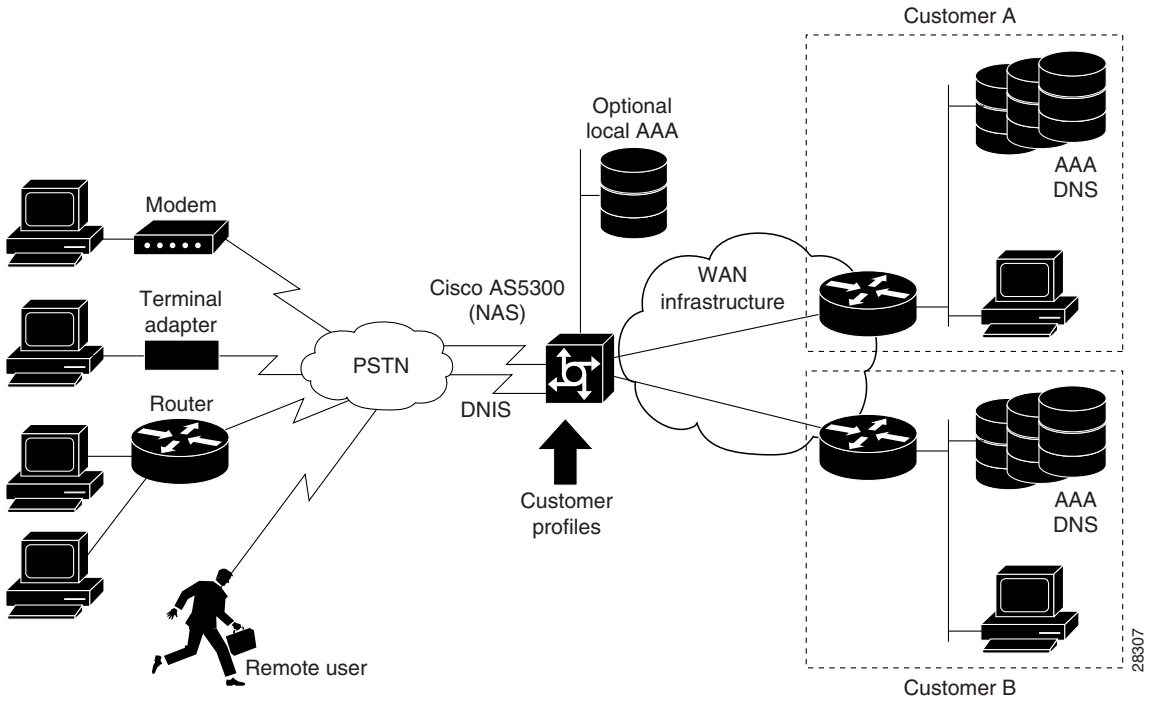


Figure 40 shows a method of implementing wholesale dial services without using VPDN tunnels by creating individual customer profiles that consist of AAA groups and PPP configurations. The AAA groups provide IP addresses of AAA servers for authentication and accounting. The PPP configurations enable you to set different PPP parameter values on each customer profile. A customer profile typically includes the following PPP parameters:

- Applicable IP address pools or a default local list of IP addresses
- Primary and secondary DNS or WINS
- Authentication method such as the Password Authentication Protocol (PAP), Challenge Handshake Authentication Protocol (CHAP), or Microsoft CHAP Version 1 (MS-CHAP)
- Number of links allowed for each call using Multilink PPP

Note The AAA and PPP integration applies to a single NAS environment; the external RPMS solution is not supported.

Figure 40 Resource Pool Management with Direct Remote Services



Call Processing

For call processing, incoming calls are matched to a DNIS group and the customer profile associated with that DNIS group. If a match is found, the customer profile session and overflow limits are applied and if available, the required resources are allocated. If a DNIS group is not found, the customer profile associated with the default DNIS group is used. The call is rejected if a customer profile using the default DNIS group cannot be found.

After the call is answered and if VPDN is enabled, the Cisco RPM checks the customer profile for an assigned VPDN group or profile. If a VPDN group is found, Cisco RPM authorizes VPDN by matching the group domain name or DNIS with the incoming call. If a match is found, VPDN profile session and overflow limits are applied, and, if the limits are not exceeded, tunnel negotiation begins. If the VPDN limits are exceeded, the call is disconnected.

If no VPDN profile is assigned to the customer profile and VPDN is enabled, non-RPM VPDN service will be attempted. If it fails, the call is processed as a retail dial service call if local AAA service is available.

Base Session and Overflow Session Limits

Cisco RPM enables you to set base and overflow session limits in each customer profile. The base session limit determines the maximum number of nonoverflow sessions supported for a customer profile. When the session limit is reached, if overflow sessions are not enabled, any new calls are rejected. If overflow sessions are enabled, new sessions up to the session overflow limit are processed and marked as overflow for call handling and accounting.

The session overflow limit determines the allowable number of sessions above the session limit. If the session overflow limit is greater than zero, overflow sessions are enabled and the maximum number of allowed sessions is the session limit plus the session overflow limit. While the session overflow limit has been reached, any new calls are rejected. Table 12 summarizes the effects of session and session overflow limits.

Enabling overflow sessions is useful for allocating extra sessions for preferred customers at premium rates. Overflow sessions can also be useful for encouraging customers to adequately forecast bandwidth usage or for special events when normal session usage is exceeded. For example, if a customer is having a corporate-wide program and many people are expected to request remote access, you could enable many overflow sessions and charge a premium rate for the excess bandwidth requirements.

Note An overflow call is a call received while the session limit is exceeded and is in an overflow state. When a call is identified as an overflow call, the call maintains the overflow status throughout its duration, even if the number of current sessions returns below the session limit.

Table 12 Effects of Session Limit and Session Overflow Limit Settings Combinations

Base Session Limit	Session Overflow Limit	Call Handling
0	0	Reject all calls.
10	0	Accept up to 10 sessions.
10	10	Accept up to 20 sessions and mark sessions 11 to 20 as overflow sessions.
0	10	Accept up to 10 sessions and mark sessions 1 to 10 as overflow.
All	0	Accept all calls.
0	All	Accept all calls and mark all calls as overflow.

VPDN Session and Overflow Session Limits

Cisco RPM enables you to configure base and overflow session limits per VPDN profile for managing VPDN sessions.

Note The VDPN session and session overflow limits are independent of the limits set in the customer profiles.

The base VPDN session limit determines the maximum number of nonoverflow sessions supported for a VPDN profile. When the VPDN session limit is reached, if overflow sessions are not enabled, any new VPDN calls using the VPDN profile sessions are rejected. If overflow sessions are enabled, new sessions up to the session overflow limit are processed and marked as overflow for VPDN accounting.

The VPDN session overflow limit determines the number of sessions above the session limit allowed in the VPDN group. If the session overflow limit is greater than zero, overflow sessions are enabled and the maximum number of allowed sessions is the session limit plus the session overflow limit. While the session overflow limit has been reached, any new calls are rejected.

Enabling VPDN overflow sessions is useful for allocating extra sessions for preferred customers at premium rates. Overflow sessions are also useful for encouraging customers to adequately forecast bandwidth usage or for special events when normal session usage is exceeded. For example, if a

customer is having a corporate-wide program and many people are expected to request remote access, you could enable many overflow sessions and charge a premium rate for the extra bandwidth requirements.

VPDN MLP Bundle and Links-per-Bundle Limits

To ensure that resources are not consumed by a few users with MLP connections, Cisco RPM also enables you to specify the maximum number of MLP bundles that can open in a VPDN group. In addition, you can specify the maximum number of links for each MLP bundle.

For example, if standard ISDN users access the VPDN profile, limit this setting to two links per bundle. If video conferencing is used, increase this setting to accommodate the necessary bandwidth (usually six links). These limits have no overflow option and are configured under the VPDN group component.

VPDN Tunnel Limits

For increased VPDN tunnel management, Cisco RPM enables you to set an IP endpoint session limit for each IP endpoint. IP endpoints are configured for VPDN groups.

Figure 41 and Figure 42 show logical flowcharts of RPM call processing for a standalone NAS with and without the RPM Direct Remote Services feature.

Figure 41 RPM Call-Processing Flowchart for a Standalone Network Access Server

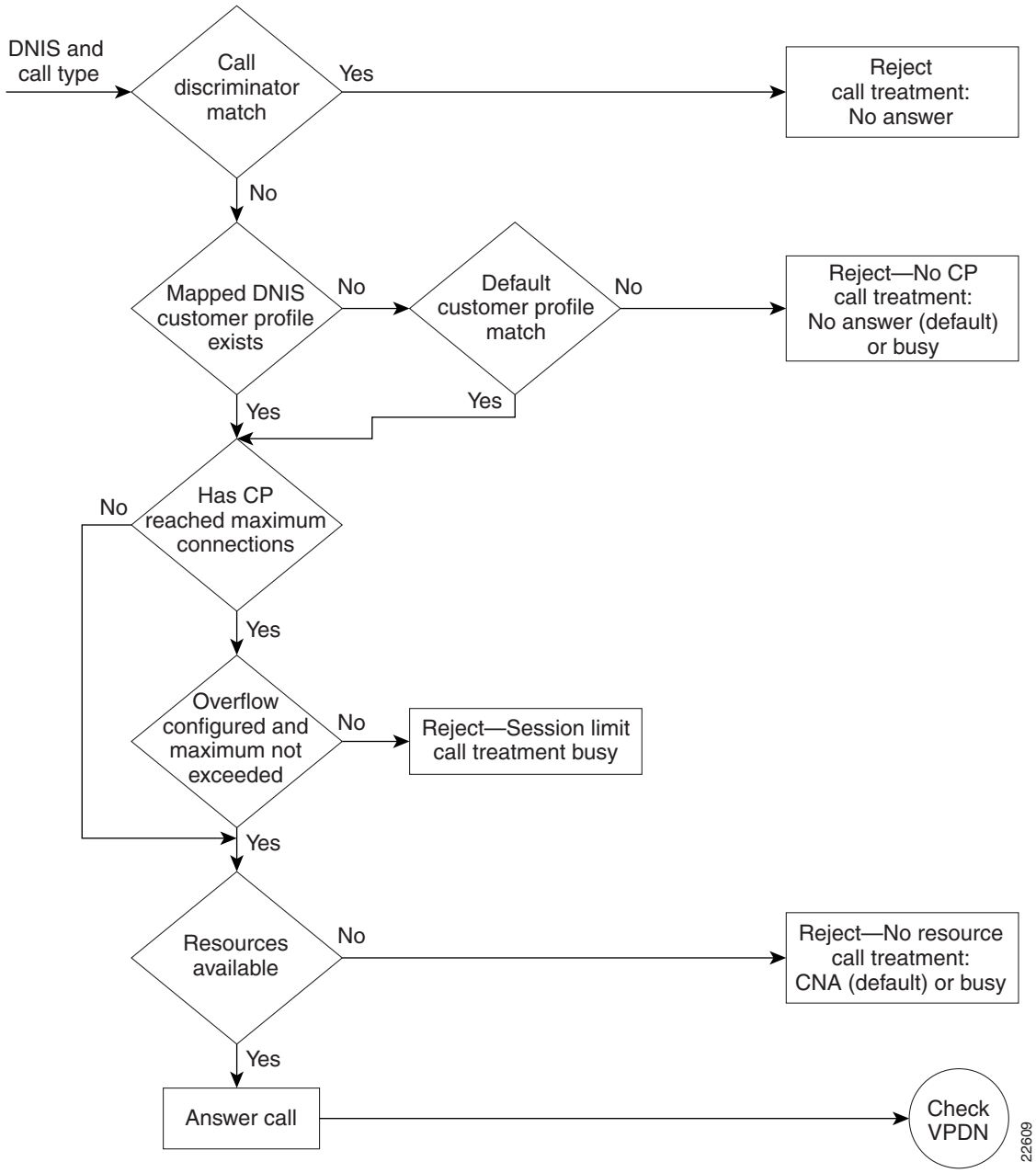
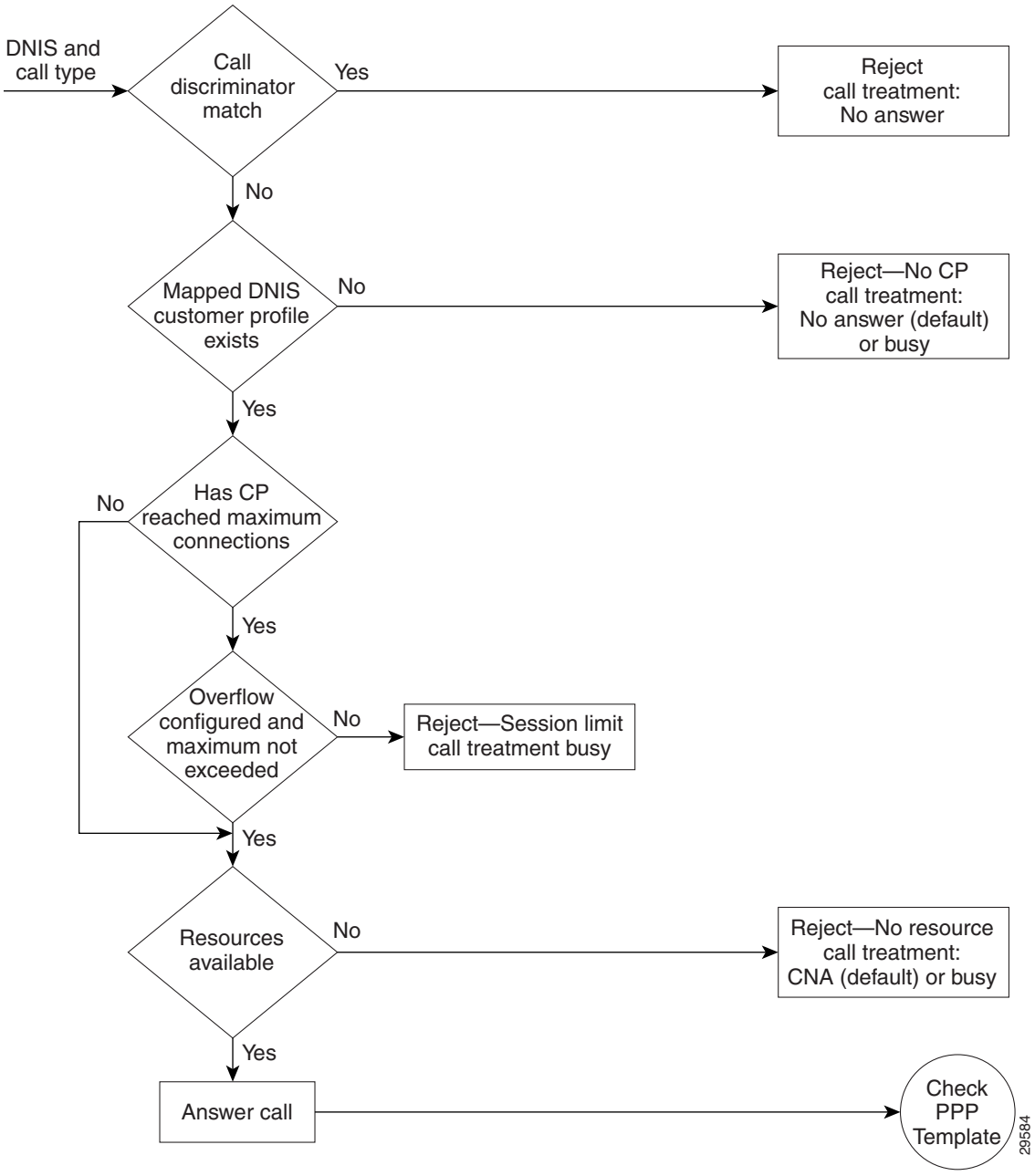


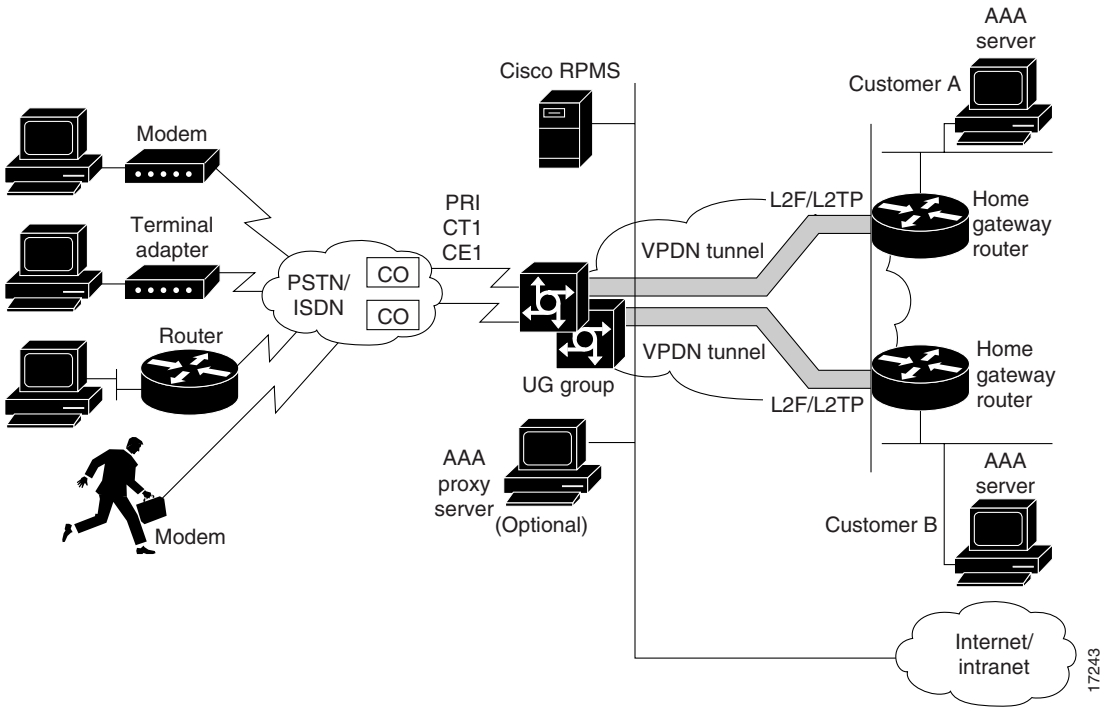
Figure 42 Flowchart for a Standalone Network Access Server with RPM Direct Remote Services



RPM Using the Cisco RPMS

Figure 43 shows a typical resource pooling network scenario using RPMS.

Figure 43 RPM Scenario Using RPMS

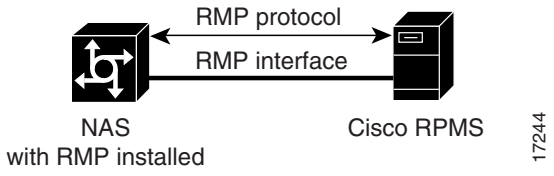


Resource Manager Protocol

Resource Manager Protocol (RMP) is a robust, recoverable protocol used for communication between the Cisco RPMS and the NAS. Each NAS client uses RMP to communicate resource management requests to the Cisco RPMS server. RPMS also periodically polls the NAS clients to query their current call information or address error conditions when they occur. RMP also allows for protocol attributes that make it extensible and enable support for customer billing requirements.

Figure 44 shows the relationship of Cisco RPM CLID/DNIS Call Discriminator Feature and RMP.

Figure 44 Cisco RPM CLID/DNIS Call Discriminator Feature and RMP



Note RMP must be enabled on all NASes that communicate with the Cisco RPM CLID/DNIS Call Discriminator Feature.

Direct Remote Services

Direct remote services is an enhancement to Cisco RPM implemented in Cisco IOS Release 12.0(7)T that enables service providers to implement wholesale dial services without using VPDN tunnels. A customer profile that has been preconfigured with a PPP template to define the unique PPP services for the wholesale dial customer is selected by the incoming DNIS and call type. At the same time, the DNIS is used to select AAA server groups for authentication/authorization and for accounting for the customer.

PPP Common Configuration Architecture (CCA) is the new component of the RPM customer profile that enables direct remote services. The full PPP command set available in Cisco IOS software is configurable per customer profile for wholesale dial applications. A customer profile typically includes the following PPP parameters:

- Local or named IP address pools
- Primary and secondary DNS or WINS addresses
- Authentication method (PAP, CHAP, MS-CHAP)
- Multilink PPP links per bundle limits

The AAA session information is selected by the incoming DNIS. AAA server lists provide the IP addresses of AAA servers for authentication, authorization, and accounting in the wholesale local network of the customer. The server lists for both authentication and authorization and for accounting contain the server addresses, AAA server type, timeout, retransmission, and keys per server.

When direct remote services is implemented on a Cisco NAS, the following sequence occurs:

1. The NAS sends an authorization request packet to the AAA server by using the authentication method (PAP, CHAP, MSCHAP) that has been configured through PPP.
2. The AAA server accepts the authorization request and returns one of the following items to the NAS:
   - A specific IP address
   - An IP address pool name
   - Nothing
3. Depending on the response from the AAA server, the NAS assigns one of the following items to the user through the DNS/WINS:
   - The IP address returned by the AAA server
   - An IP address randomly assigned from the named IP address pool
   - An IP address from a pool specified in the customer profile template

Note If the AAA server sends back to the NAS a named IP address pool and that name does not exist on the NAS, the request for service is denied. If the AAA server does not send anything back to the NAS and there is an IP address pool name configured in the customer profile template, an address from that pool is used for the session.

RPM Process with RPMS and SS7

For information on SS7 implementation for RPM, refer to the document *Cisco Resource Pool Manager Server 1.0 SS7 Implementation.*

Additional Information About Cisco RPM

For more information about Cisco RPM, see the following documents:

- *AAA Server Group*
- *Cisco Access VPN Solutions Using Tunneling Technology*
- *Cisco AS5200 Universal Access Server Software Configuration Guide*
- *Cisco AS5300 Software Configuration Guide*
- *Cisco AS5800 Access Server Software ICG*
- *Cisco Resource Pool Manager Server Configuration Guide*
- *Cisco Resource Pool Manager Server Installation Guide*
- *Cisco Resource Pool Manager Server Solutions Guide*
- *Dial Solutions Quick Configuration Guide*
- *RADIUS Multiple UDP Ports Support*
- *Redundant Link Manager*
- *Release Notes for Cisco Resource Pool Manager Server* Release 1.0
- Resource Pool Management
- Resource Pool Management with Direct Remote Services
- *Resource Pool Manager Customer Profile Template*
- *Selecting AAA Server Groups Based on DNIS*
- *SS7 Continuity Testing for Network Access Servers*
- *SS7 Dial Solution System Integration*

How to Configure RPM

Read and comply with the following restrictions and prerequisites before beginning RPM configuration:

- RPM is supported on Cisco AS5300, Cisco AS5400, and Cisco AS5800 Universal Access Servers
- Modem pooling and RPM are not compatible.
- The Cisco RPM CLID/DNIS Call Discriminator Feature must have Cisco RPM configured.
- CLID screening is not available to channel-associated signaling (CAS) interrupt level calls.
- Cisco RPM requires the NPE 300 processor when implemented on the Cisco AS5800.
- For Cisco AS5200 and Cisco AS5300 access servers, Cisco IOS Release 12.0(4)XI1 or later releases must be running on the NAS.
- For Cisco AS5800, Cisco IOS Release 12.0(5)T or later releases must be running on the NAS.
- A minimum of 64 MB must be available on the DMM cards.
- The RPM application requires an NPE 300.
- For call discriminator profiles, the Cisco AS5300, Cisco AS5400, or Cisco AS5800 Universal Access Servers require a minimum of 16 MB Flash memory and 128 MB DRAM memory, and need to be configured for VoIP as an H.323-compliant gateway.

The following tasks must be performed before configuring RPM:

- Accomplish initial configuration as described in the appropriate *Universal Access Server Software Configuration Guide*. Perform the following tasks as required.
  - Set your local AAA
  - Define your TACACS+ server for RPM
  - Define AAA accounting
  - Ensure PPP connectivity
  - Ensure VPDN connectivity

  Refer to the document *Configuring the NAS for Basic Dial Access* for more information.

To configure your NAS for RPM, perform the following tasks:

- Enabling RPM (Required)
- Configuring DNIS Groups (As required)
- Creating CLID Groups (As required)
- Configuring Discriminator Profiles (As required)
- Configuring Resource Groups (As required)
- Configuring Service Profiles (As required)
- Configuring Customer Profiles (As required)
- Configuring a Customer Profile Template (As required)
- Placing the Template in the Customer Profile (As required)
- Configuring AAA Server Groups (As required)
- Configuring VPDN Profiles (As required)
- Configuring VPDN Groups (As required)
- Counting VPDN Sessions by Using VPDN Profiles (As required)
- Limiting the Number of MLP Bundles in VPDN Groups (As required)
- Configuring Switched 56 over CT1 and RBS (As required)

See the section "Troubleshooting RPM" later in this chapter for troubleshooting tips. See the section "Configuration Examples for RPM" at the end of this chapter for examples of how to configure RPM in your network.

Enabling RPM

To enable RPM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool enable`	Turns on RPM.
Step 2	`Router(config)# resource-pool call treatment resource channel-not-available`	Creates a resource group for resource management.
Step 3	`Router(config)# resource-pool call treatment profile no-answer`	Sets up the signal sent back to the telco switch in response to incoming calls.
Step 4	`Router(config) # resource-pool aaa protocol local`	Specifies which protocol to use for resource management.

Note If you have an RPMS, you need not define VPDN groups/profiles, customer profiles, or DNIS groups on the NAS; you need only define resource groups. Configure the remaining items by using the RPMS system.

Configuring DNIS Groups

This configuration task is optional.

To configure DNIS groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dialer dnis group` *dnis-group-name*	Creates a DNIS group. The name you specify in this step must match the name entered when configuring the customer profile.
Step 2	`Router(config-called-group)# call-type cas {digital | speech}`	Statically sets the call-type override for incoming CAS calls.
Step 3	`Router(config-called-group)# number` *number*	Enters DNIS numbers to be used in the customer profile. (Wildcards can be used.)

For default DNIS service, no DNIS group configuration is required. The following characteristics and restrictions apply to DNIS group configuration:

- Each DNIS group/call-type combination can apply to only one customer profile.
- You can use up to four default DNIS groups (one for each call type).
- You must statically configure CAS call types.
- You can use x, X or . as wildcards within each DNIS number.

Creating CLID Groups

You can add multiple CLID groups to a discriminator profile. You can organize CLID numbers for a customer or service type into a CLID group. Add all CLID numbers into one CLID group, or subdivide the CLID numbers using criteria such as call type, geographical location, or division. To create CLID groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# dialer clid group clid-group-name`	Creates a CLID group, assigns it a name of up to 23 characters, and enters CLID configuration mode. The CLID group must be the same as the group specified in the customer profile configuration. Refer to the *Resource Pool Management with Direct Remote Services* document for information on configuring customer profiles.
Step 2	`Router(config-clid-group)# number clid-group-number`	Enters CLID configuration mode, and adds a CLID number to the dialer CLID group that is used in the customer profile. The CLID number can have up to 65 characters. You can use **x**, **X** or **.** as wildcards within each CLID number. The CLID screening feature rejects this number if it matches the CLID of an incoming call.

Configuring Discriminator Profiles

Discriminator profiles enable you to process calls differently on the basis of the call type and CLID/DNIS combination. The "Call Discriminator Profiles" section earlier in this chapter describes the different types of discriminator profiles that you can create.

To configure discriminator profiles for RPM implementation, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile discriminator name`	Creates a call discriminator profile and assigns it a name of up to 23 characters.
Step 2	`Router(config-call-d)# call-type {all | digital | speech | v110 | v120}`	Specifies the type of calls you want to block. The NAS will not answer the call-type you specify.

	Command	Purpose
Step 3	`Router(config-call-d)# clid group {clid-group-name | default}`	Optional. Associates a CLID group with the discriminator. If you do not specify a *clid-group-nam*e, the default discriminator in the RM is used. Any CLID number coming in on a call is in its respective default group unless it is specifically assigned a *clid-group-nam*e. After a CLID group is associated with a call type in a discriminator, it cannot be used in any other discriminator.
Step 4	`Router(config-call-d)# dnis group {dnis-group-name | default}`	Optional. Associates a DNIS group with the discriminator. If you do not specify a *dnis-group-nam*e, the default discriminator in the RM is used. Any DNIS number coming in on a call is in its respective default group unless it is specifically assigned a *dnis-group-nam*e. After a DNIS group is associated with a call type in a discriminator, it cannot be used in any other discriminator.

To verify discriminator profile settings, use the following commands:

Step 1 Use the **show resource-pool discriminator** *name* command to verify the call discriminator profiles that you configured.

If you enter the **show resource-pool discriminator** command without including a call discriminator name, a list of all current call discriminator profiles appears.

If you enter a call discriminator profile *name* with the **show resource-pool discriminator** command, the number of calls rejected by the selected call discriminator appears.

```
Router# show resource-pool discriminator

List of Call Discriminator Profiles:
    deny_CLID

Router# show resource-pool discriminator deny_CLID

    1 calls rejected
```

Step 2 Use the **show dialer** command to display general diagnostic information for interfaces configured for the dialer.

```
Router# show dialer [interface] type number
```

Configuring Resource Groups

To configure resource groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool group resource name`	Creates a resource group and assign it a name of up to 23 characters.
Step 2	`Router(config-resource-group)# range {port {slot/port slot/port}} | {limit number}`	Associates a range of modems or other physical resources with this resource group: • For port-based resources, use the physical locations of the resources. • For non-port-based resources, use a single integer limit. Specify the maximum number of simultaneous connections supported by the resource group. Up to 192 connections may be supported, depending on the hardware configuration of the access server.

For external Cisco RPMS environments, configure resource groups on the NAS before defining them on external RPMS servers.

For standalone NAS environments, first configure resource groups before using them in customer profiles.

Resource groups can apply to multiple customer profiles.

Note You can separate physical resources into groups. However, do not put heterogeneous resources in the same group. Do not put MICA technologies modems in the same group as Microcom modems. Do not put modems and HDLC controllers in the same resource group. Do not configure the **port** and **limit** command parameters in the same resource group.

Configuring Service Profiles

To configure service profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile service name`	Creates a service profile and assign it a name of up to 23 characters.
Step 2	`Router(config-service-profil)# modem min-speed {speed | any} max-speed {speed | any [modulation value]}`	Specifies the desired modem parameter values. The range for **min-speed** and **max-speed** is 300 to 56000 bits per second.

Service profiles are used to configure modem service parameters for Nextport and MICA technologies modems, and support speech, digital, V.110, and V.120 call types. Error-correction and compression are hidden parameters that may be included in a service profile.

Configuring Customer Profiles

To configure customer profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Creates a customer profile.
Step 2	`Router(config-customer-pro)# dnis group` {*dnis-group-name* | **default**}	Includes a group of DNIS numbers in the customer profile.
Step 3	`Router(config-customer-pro)# limit base-size` {*number* | **all**}	Specifies the base size usage limit.
Step 4	`Router(config-customer-pro)# limit overflow-size` {*number* | **all**}	Specifies the oversize size usage limit.
Step 5	`Router(config-customer-pro)# resource` *WORD* {**digital** | **speech** | **v110** | **v120**} [**service** *WORD*]	Assigns resources and supported call types to the customer profile.

Customer profiles are used so that service providers can assign different service characteristics to different customers. Note the following characteristics of customer profiles:

- Multiple resources of the same call type are used sequentially.
- The limits imposed are per customer (DNIS)—not per resource.
- A digital resource with a call type of **speech** allows for Data over Speech Bearer Service (DoSBS).

Configuring Default Customer Profiles

Default customer profiles are identical to standard customer profiles, except they do not have any associated DNIS groups. To define a default customer profile, use the reserved keyword **default** for the DNIS group:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to the default customer profile.
Step 2	`Router(config-customer-pro)# dnis group default`	Assigns the default DNIS group to the customer profile. This sets up the customer profile such that it will use the default DNIS configuration, which is automatically set on the NAS.

The rest of the customer profile is configured as shown in the previous section "Configuring Customer Profiles."

Configuring Customer Profiles Using Backup Customer Profiles

Backup customer profiles are customer profiles configured locally on the Cisco NAS and are used to answer calls on the basis of a configured allocation scheme when the link between the Cisco NAS and Cisco RPMS is disabled.

To enable the backup feature, you need to have already configured the following on the router:

- The **resource-pool aaa protocol group** *name* **local** command.
- All customer profiles and DNIS groups on the NAS.

The backup customer profile can contain all of the elements defined in a standard customer profile, including base size or overflow parameters. However, when the connection between the Cisco NAS and Cisco RPMS is unavailable, session counting and session limits are not applied to incoming calls. Also, after the connection is reestablished, there is no synchronization of call counters between the Cisco NAS and Cisco RPMS.

Configuring Customer Profiles for Using DoVBS

To configure customer profiles for using DoVBS, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to a customer profile.
Step 2	`Router(config-customer-pro)# dnis group` *name*	Assigns a DNIS group to the customer profile. DNIS numbers are assigned as shown in the previous section.
Step 3	`Router(config)# limit base-size` {*number* | **all**}	Specifies the VPDN base size usage limit.
Step 4	`Router(config)# limit overflow-size` {*number* | **all**}	Specifies the VPDN overflow size usage limit.
Step 5	`Router(config-customer-pro)# resource` *name* {**digital** | **speech** | **v110** | **v120**} [**service** *name*]	Specifies resource names to use within the customer profile.

To support ISDN DoVBS, use a DNIS group and a configured customer profile to direct the speech call to the appropriate digital resource. The DNIS group assigned to the customer profile should have a call type of speech. The resource group assigned to this customer profile will be digital resources and also have a call type of speech, so the call will terminate on an HDLC controller rather than a modem.

See the section "Customer Profile Configuration for DoVBS Example" at the end of this chapter for a configuration example.

Configuring a Customer Profile Template

Customer profile templates provide a way to keep each unique situation for a customer separate for both security and accountability. This is an optional configuration task.

To configure a template and place it in a customer profile, ensure that all basic configuration tasks and the RPM configuration tasks have been completed and verified before attempting to configure the customer profile templates.

To add PPP configurations to a customer profile, create a customer profile template. Once you create the template and associate it with a customer profile by using the **source template** command, it is integrated into the customer profile.

To configure a template in RPM, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# template name`	Creates a customer profile template and assign a unique name that relates to the customer that will be receiving it. **Note** Steps 2, 3, and 4 are optional. Enter multilink, peer, and ppp commands appropriate to the application requirements of the customer.
Step 2	`Router(config-template)# peer default ip address pool pool-name`	(Optional) Specifies that the customer profile to which this template is attached will use a local IP address pool with the specified name.
Step 3	`Router(config-template)# ppp authentication chap`	(Optional) Sets the PPP link authentication method.
Step 4	`Router(config-template)# ppp multilink`	(Optional) Enables Multilink PPP for this customer profile.
Step 5	`Router(config-template)# exit`	Exits from template configuration mode; returns to global configuration mode.
Step 6	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this template.
Step 7	`Router(config-customer-profi)# source template name`	Attaches the customer profile template you have just configured to the customer profile.

Typical Template Configuration

The following example shows a typical template configuration:

```
template Word
  multilink {max-fragments frag-num | max-links num | min-links num}
  peer match aaa-pools
  peer default ip address {pool pool-name1 [pool-name2] | dhcp}
  ppp ipcp {dns | wins} A.B.C.D [W.X.Y.Z]
resource-pool profile customer WORD
  source template Word
  aaa group-configuration aaa-group-name

template acme_direct
  peer default ip address pool tahoe
  ppp authentication chap isdn-users
  ppp multilink
```

Verifying Template Configuration

To verify your template configuration, perform the following steps:

Step 1 Enter the **show running-config** EXEC command (where the template name is "PPP1"):

```
Router#
Router# show running-config begin template
.
.
.
template PPP1
peer default ip address pool pool1 pool2
```

```
ppp ipcp dns 10.1.1.1 10.1.1.2
ppp ipcp wins 10.1.1.3 10.1.1.4
ppp multilink max-links 2
.
.
.
```

Step 2 Ensure that your template appears in the configuration file.

Placing the Template in the Customer Profile

To place your template in the customer profile, use the following commands beginning in global configuration command mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile customer` *name*	Assigns a name to a customer profile.
Step 2	`Router(config-customer-pr)# source template`	Associates the template with the customer profile.

To verify the placement of your template in the customer profile, perform the following steps:

Step 1 Enter the **show resource-pool customer** EXEC command:

```
Router# show resource-pool customer

List of Customer Profiles:
    CP1
    CP2
```

Step 2 Look at the list of customer profiles and make sure that your profile appears in the list.

Step 3 To verify a particular customer profile configuration, enter the **show resource-pool customer** *name* EXEC command (where the customer profile name is "CP1"):

```
Router# show resource-pool customer CP1

97 active connections
    120 calls accepted
    210 max number of simultaneous connections
    50 calls rejected due to profile limits
    0 calls rejected due to resource unavailable
    90 minutes spent with max connections
    5 overflow connections
    2 overflow states entered
    0 overflow connections rejected
    0 minutes spent in overflow
    13134 minutes since last clear command
```

Configuring AAA Server Groups

To configure AAA server groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# aaa new-model`	Enables AAA on the NAS.

Step 2	`Router(config)# radius-server key key`
	or
	`Router(config)# tacacs-server key key`
		Set the authentication and encryption key used for all RADIUS or TACACS+ communications between the NAS and the RADIUS or TACACS+ daemon.

Step 3	`Router(config)# radius-server host {hostname | ip-address key} [auth-port port acct-port port]`
	or
	`Router(config)# tacacs-server host ip-address key`
		Specifies the host name or IP address of the server host before configuring the AAA server group. You can also specify the UDP destination ports for authentication and for accounting.

Step 4	`Router(config)# aaa group server {radius | tacacs+} group-name`	Selects the AAA server type you want to place into a server group and assign a server group name.
Step 5	`Router(config-sg radius)# server ip-address`	Specifies the IP address of the selected server type. This must be the same IP address that was assigned to the server host in Step 3.
Step 6	`Router(config-sg radius)# exit`	Returns to global configuration mode.
Step 7	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this AAA server group.
Step 8	`Router(config-customer-profil)# aaa group-configuration group-name`	Associates this AAA server group (named in Step 4) with the customer profile named in Step 7.

AAA server groups are lists of AAA server hosts of a particular type. The Cisco RPM currently supports RADIUS and TACACS+ server hosts. A AAA server group lists the IP addresses of the selected server hosts.

You can use a AAA server group to define a distinct list of AAA server hosts and apply this list to the Cisco RPM application. Note that the AAA server group feature works only when the server hosts in a group are of the same type.

Configuring VPDN Profiles

A VPDN profile is required only if you want to impose limits on the VPDN tunnel that are separate from the customer limits.

To configure VPDN profiles, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile vpdn profile-name`	Creates a VPDN profile and assigns it a profile name
Step 2	`Router(config-vpdn-profile)# limit base-size {number | all}`	Specifies the maximum number of simultaneous base VPDN sessions to be allowed for this VPDN group under the terms of the service-level agreement (SLA). The range is 0 to 1000 sessions. If all sessions are to be designated as base VPDN sessions, specify **all**.

	Command	Purpose
Step 3	`Router(config-vpdn-profile)# limit overflow-size {number | all}`	Specifies the maximum number of simultaneous overflow VPDN sessions to be allowed for this VPDN group under the terms of the SLA. The range is 0 to 1000 sessions. If all sessions are to be designated as overflow VPDN sessions, specify **all**.
Step 4	`Router(config-vpdn-profile)# exit`	Returns to global configuration mode.
Step 5	`Router(config)# resource-pool profile customer name`	Enters customer profile configuration mode for the customer to which you wish to assign this VPDN group.

Step 6	`Router(config-customer-profi)# vpdn profile profile-name`
	or
	`Router(config-customer-profi)# vpdn group group-name`
		Attaches the VPDN profile you have just configured to the customer profile to which it belongs, or, if the limits imposed by the VPDN profile are not required, attaches VPDN group instead (see the section "Configuring VPDN Groups" later in this chapter).

Configuring VPDN Groups

To configure VPDN groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn enable`	Enables VPDN sessions on the NAS.
Step 2	`Router(config)# vpdn-group group-name`	Creates a VPDN group and assigns it a unique name. Each VPDN group can have multiple endpoints (HGW/LNSs).
Step 3	`Router(config-vpdn)# request dialin {l2f | l2tp} {ip ip-address} {domain domain-name | dnis dnis-number}`	Specifies the tunneling protocol to be used to reach the remote peer defined by a specific IP address if a dial-in request is received for the specified domain name or DNIS number. The IP address that qualifies the session is automatically generated and need not be entered again. **Note** Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.
Step 4	`Router(config-vpdn)# multilink {bundle-number | link-number}`	Specifies the maximum number of bundles and links for all multilink users in the VPDN group. The range for both bundles and links is 0 to 32767. In general, each user requires one bundle.
Step 5	`Router(config-vpdn)# loadsharing ip ip-address [limit number]`	Configures the endpoints for loadsharing. This router will share the load of IP traffic with the first router specified in Step 2. The **limit** keyword limits the number of simultaneous sessions that are sent to the remote endpoint (HGW/LNS). This limit can be 0 to 32767 sessions.

	Command	Purpose
Step 6	`Router(config-vpdn)# backup ip ip-address [limit number] [priority number]`	Sets up a backup HGW/LNS router. The number of sessions per backup can be limited. The priority number can be 2 to 32767. The highest priority is 2, which is the first HGW/LNS router to receive backup traffic. The lowest priority, which is the default, is 32767.
Step 7	`Router(config-vpdn)# exit`	Returns to global configuration mode.

Step 8	`Router(config)# resource-pool profile vpdn profile-name`
	or
	`Router(config)# resource-pool profile customer name`
		Enters either VPDN profile configuration mode or customer profile configuration mode, depending on whether you want to allow VPDN connections for a customer profile, or allow combined session counting on all of the VPDN sessions within a VPDN profile.

Step 9	`Router(config-vpdn-profile)# vpdn group group-name`
	or
	`Router(config-customer-profi)# vpdn group group-name`
		Attaches the VPDN group to either the VPDN profile or the customer profile specified in Step 8.

A VPDN group consists of VPDN sessions that are combined and placed into a customer profile or a VPDN profile. Note the following characteristics of VPDN groups:

- The *dnis-group-name* argument is required to authorize the VPDN group with RPM.
- A VPDN group placed in a customer profile allows VPDN connections for the customer using that profile.
- A VPDN group placed in a VPDN profile allows the session limits configured for that profile to apply to all of the VPDN sessions within that VPDN group.
- VPDN data includes an associated domain name or DNIS, an endpoint IP address, the maximum number of MLP bundles, and the maximum number of links per MLP bundle; this data can optionally be located on a AAA server.

See the sections "VPDN Configuration Example" and "VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example" at the end of this chapter for examples of using VPDN with RPM.

Counting VPDN Sessions by Using VPDN Profiles

Session counting is provided for each VPDN profile. One session is brought up each time a remote client dials into a HGW/LNS router by using the NAS/LAC. Sessions are counted by using VPDN profiles. If you do not want to count the number of VPDN sessions, do not set up any VPDN profiles. VPDN profiles count sessions in one or more VPDN groups.

To configure VPDN profile session counting, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# resource-pool profile vpdn` *name*	Creates a VPDN profile.
Step 2	`Router(config-vpdn-profile)# vpdn-group` *name* `Router(config-vpdn-profile)# exit`	Associates a VPDN group to the VPDN profile. VPDN sessions done within this VPDN group will be counted by the VPDN profile.
Step 3	`Router(config)# resource-pool profile customer` *name* `Router(config-customer-profi)# vpdn profile` *name*	Links the VPDN group to a customer profile.
Step 4	`Router(config-customer-profi)# ^Z` `Router#`	Returns to EXEC mode to perform verification steps.

To verify session counting and view VPDN group information configured under resource pooling, use the **show resource-pool vpdn group** command. In this example, two different VPDN groups are configured under two different customer profiles:

```
Router# show resource-pool vpdn group

List of VPDN Groups under Customer Profiles
Customer Profile customer1:customer1-vpdng
Customer Profile customer2:customer2-vpdng
List of VPDN Groups under VPDN Profiles
VPDN Profile customer1-profile:customer1-vpdng
```

To display the contents of a specific VPDN group, use the **show resource-pool vpdn group** *name* command. This example contains one domain name, two DNIS called groups, and two endpoints:

```
Router# show resource-pool vpdn group customer2-vpdng

VPDN Group customer2-vpdng found under Customer Profiles: customer2

Tunnel (L2TP)
------
dnis:cg1
dnis:cg2
dnis:jan

Endpoint        Session Limit Priority Active Sessions Status Reserved Sessions
--------        ------------- -------- --------------- ------ -----------------
172.21.9.67     *             1        0               OK     -
10.1.1.1         *             2        0               OK     -
--------------- ------------- -------- --------------- ------ -----------------
Total           *                      0                      0
```

To display the contents of a specific VPDN profile, use the **show resource-pool vpdn profile** *name* command, as follows:

```
Router# show resource-pool vpdn profile ?

  WORD  VPDN profile name
  <cr>

Router# show resource-pool vpdn profile customer1-profile

    0 active connections
    0 max number of simultaneous connections
    0 calls rejected due to profile limits
    0 calls rejected due to resource unavailable
    0 overflow connections
    0 overflow states entered
```

```
    0 overflow connections rejected
    1435 minutes since last clear command
```

Note Use the **debug vpdn event** command to troubleshoot VPDN profile limits, session limits, and MLP connections. First, enable this command; then, send a call into the access server. Interpret the debug output and make configuration changes as needed.

To debug the L2F or L2TP protocols, use the **debug vpdn l2x** command:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
Router# debug vpdn l2x ?

  error            VPDN Protocol errors
  event            VPDN event
  l2tp-sequencing  L2TP sequencing
  l2x-data         L2F/L2TP data packets
  l2x-errors       L2F/L2TP protocol errors
  l2x-events       L2F/L2TP protocol events
  l2x-packets      L2F/L2TP control packets
  packet           VPDN packet
```

Limiting the Number of MLP Bundles in VPDN Groups

Cisco IOS software enables you to limit the number of MLP bundles and links supported for each VPDN group. A bundle name consists of a username endpoint discriminator (for example, an IP address or phone number) sent during LCP negotiation.

To limit the number of MLP bundles in VPDN groups, use the following commands beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# vpdn-group` *name*	Creates a VPDN group.
Step 2	`Router(config-vpdn)# multilink {bundle` *number* `| link` *number*`}`	Limits the number of MLP bundles per VPDN group and links per bundle.[1] These settings limit the number of users that can multilink.

1. Both the NAS/LAC and the HGW/LNS router must be configured to support multilink before a client can use multilink to connect to a HGW/LNS.

The following example shows the **show vpdn multilink** command output for verifying MLP bundle limits:

```
Router# show vpdn multilink

Multilink Bundle Name   VPDN Group Active links Reserved links Bundle/Link Limit
---------------------   ---------- ------------ -------------- -----------------
twv@anycompany.com      vgdnis     0            0              */*
```

Note Use the **debug vpdn event** and **debug resource-pooling** commands to troubleshoot VPDN profile limits, session limits, and MLP connections. First, enable this command; then, send a call into the access server. Interpret the debug output and make configuration changes as needed.

Configuring Switched 56 over CT1 and RBS

To configure switched 56 over CT1 and RBS, use the following commands beginning in global configuration mode. Perform this task on the Cisco AS5200 and Cisco AS5300 access servers only.

	Command	Purpose
Step 1	`Router(config)# controller t1 number`	Specifies a controller and begins controller configuration mode.
Step 2	`Router(config-controller)# cas-group 0 timeslots 1-24 type e&m-fgb {dtmf | mf} {dnis}`	Creates a CAS group and assigns time slots.
Step 3	`Router(config-controller)# framing {sf | esf}`	Specifies framing.
Step 4	`Router(config-controller)# linecode {ami | b8zs}`	Enters the line code.
Step 5	`Router(config-controller)# exit`	Returns to global configuration mode.
Step 6	`Router(config)# dialer dnis group name`	Creates a dialer called group.
Step 7	`Router(config-called-group)# call-type cas digital`	Assigns a call type as digital (switch 56).
Step 8	`Router(config-called-group)# exit`	Returns to global configuration mode.

Step 9	`Router(config)# interface serial number:number`
	
	`Router(config-if)#`
		Specifies the logical serial interface, which was dynamically created when the **cas-group** command was issued.
		
		This command also enters interface configuration mode, where you configure the core protocol characteristics for the serial interface.

To verify switched 56 over CT1, use the **show dialer dnis** command as follows:

```
Router# show dialer dnis group

List of DNIS Groups:
  default
  mdm_grp1

Router# show dialer dnis group mdm_grp1

Called Number:2001
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2002
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2003
  0 total connections
  0 peak connections
  0 calltype mismatches
Called Number:2004
  0 total connections
  0 peak connections
  0 calltype mismatches
.
.
.
Router# show dialer dnis number
```

```
List of Numbers:
  default
  2001
  2002
  2003
  2004
.
.
.
```

Verifying RPM Components

The following sections provide call-counter and call-detail output for the different RPM components:

- Verifying Current Calls
- Verifying Call Counters for a Customer Profile
- Clearing Call Counters
- Verifying Call Counters for a Discriminator Profile
- Verifying Call Counters for a Resource Group
- Verifying Call Counters for a DNIS Group
- Verifying Call Counters for a VPDN Profile
- Verifying Load Sharing and Backup

Verifying Current Calls

The following output from the **show resource-pool call** command shows the details for all current calls, including the customer profile and resource group, and the matched DNIS group:

```
Router# show resource-pool call

Shelf 0, slot 0, port 0, channel 15, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group isdn-ports
  DNIS number 301001
Shelf 0, slot 0, port 0, channel 14, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group isdn-ports
  DNIS number 301001
Shelf 0, slot 0, port 0, channel 11, state RM_RPM_RES_ALLOCATED
  Customer profile ACME, resource group MICA-modems
  DNIS number 301001
```

Verifying Call Counters for a Customer Profile

The following output from the **show resource-pool customer** command shows the call counters for a given customer profile. These counters include historical data and can be cleared.

```
Router# show resource-pool customer ACME

    3 active connections
    41 calls accepted
    3 max number of simultaneous connections
    11 calls rejected due to profile limits
```

```
    2 calls rejected due to resource unavailable
    0 minutes spent with max connections
    5 overflow connections
    1 overflow states entered
    11 overflow connections rejected
    10 minutes spent in overflow
    214 minutes since last clear command
```

Clearing Call Counters

The **clear resource-pool** command clears the call counters.

Verifying Call Counters for a Discriminator Profile

The following output from the **show resource-pool discriminator** command shows the call counters for a given discriminator profile. These counters include historical data and can be cleared.

```
Router# show resource-pool discriminator

List of Call Discriminator Profiles:
    deny_DNIS

Router# show resource-pool discriminator deny_DNIS

    1 calls rejected
```

Verifying Call Counters for a Resource Group

The following output from the **show resource-pool resource** command shows the call counters for a given resource group. These counters include historical data and can be cleared.

```
Router# show resource-pool resource

List of Resources:
    isdn-ports
    MICA-modems

Router# show resource-pool resource isdn-ports

    46 resources in the resource group
    2 resources currently active
    8 calls accepted in the resource group
    2 calls rejected due to resource unavailable
    0 calls rejected due to resource allocation errors
```

Verifying Call Counters for a DNIS Group

The following output from the **show dialer dnis** command shows the call counters for a given DNIS group. These counters include historical data and can be cleared.

```
Router# show dialer dnis group ACME_dnis_numbers

DNIS Number:301001
  11 total connections
  5 peak connections
```

```
  0 calltype mismatches
```

Verifying Call Counters for a VPDN Profile

The following output from the **show resource-pool vpdn** command shows the call counters for a given VPDN profile or the tunnel information for a given VPDN group. These counters include historical data and can be cleared.

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
Router# show resource-pool vpdn profile ACME_VPDN

        2 active connections
        2 max number of simultaneous connections
        0 calls rejected due to profile limits
        0 calls rejected due to resource unavailable
        0 overflow connections
        0 overflow states entered
        0 overflow connections rejected
        215 minutes since last clear command

Router# show resource-pool vpdn group outgoing-2

VPDN Group outgoing-2 found under VPDN Profiles:  ACME_VPDN

Tunnel (L2F)
------
dnis:301001
dnis:ACME_dnis_numbers

Endpoint    Session Limit Priority Active Sessions Status Reserved Sessions
--------    ------------- -------- --------------- ------ -----------------
172.16.1.9  *             1        2               OK     -
--------    -------------          ---------------        -----------------
Total       *                      2                      0
```

Verifying Load Sharing and Backup

The following example from the **show running-config** EXEC command shows two different VPDN customer groups:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
Router# show running-config

Building configuration...
.
.
.
vpdn-group customer1-vpdng
 request dialin
 protocol l2f
 domain cisco.com
 domain cisco2.com
 dnis customer1-calledg
```

```
 initiate-to ip 172.21.9.67
 loadsharing ip 172.21.9.68 limit 100
 backup ip 172.21.9.69 priority 5
vpdn-group customer2-vpdng
 request dialin
 protocol l2tp
 dnis customer2-calledg
 domain acme.com
 initiate-to ip 172.22.9.5
```

Troubleshooting RPM

Test and verify that ISDN, CAS, SS7, PPP, AAA, and VPDN are working properly before implementing RPM. Once RPM is implemented, the only **debug** commands needed for troubleshooting RPM are as follows:

- **debug resource pool**
- **debug aaa authorization**

The **debug resource-pool** command is useful as a first step to ensure proper operation. It is usually sufficient for most cases. Use the **debug aaa authorization** command for troubleshooting VPDN and modem service problems.

Problems that might typically occur are as follows:

- No DNIS group found or no customer profile uses a default DNIS
- Call discriminator blocks the DNIS
- Customer profile limits exceeded
- Resource group limits exceeded

Note Always enable the debug and log time stamps when troubleshooting RPM.

This section provides the following topics for troubleshooting RPM:

Resource-Pool Component

The resource-pool component contains two modules—a dispatcher and a local resource-pool manager. The dispatcher interfaces with the signaling stack, resource-group manager, and AAA, and is responsible for maintaining resource-pool call state and status information. The state transitions can be displayed by enabling the resource-pool debug traces. Table 13 summarizes the resource pooling states.

Table 13 Resource Pooling States

State	Description
RM_IDLE	No call activity.
RM_RES_AUTHOR	Call waiting for authorization; message sent to AAA.
RM_RES_ALLOCATING	Call authorized; resource group manager allocating.
RM_RES_ALLOCATED	Resource allocated; connection acknowledgment sent to signaling state. Call should get connected and become active.
RM_AUTH_REQ_IDLE	Signaling module disconnected call while in RM_RES_AUTHOR. Waiting for authorization response from AAA.
RM_RES_REQ_IDLE	Signaling module disconnected call while in RM_RES_ALLOCATING. Waiting for resource allocation response from resource group manager.

The resource-pool state can be used to isolate problems. For example, if a call fails authorization in the RM_RES_AUTHOR state, investigate further with AAA authorization debugs to determine whether the problem lies in the resource-pool manager, AAA, or dispatcher.

The resource-pool component also contains local customer profiles and discriminators, and is responsible for matching, configuring, and maintaining the associated counters and statistics. The resource-pool component is responsible for the following:

- Configuration of customer profiles or discriminators
- Matching a customer profile or discriminator for local profile configuration
- Counters/statistics for customer profiles or discriminators
- Active call information displayed by the **show resource-pool call** command

The RPMS debug commands are summarized in Table 14.

Table 14 Debug Commands for RPM

Command	Purpose
debug resource-pool	This debug output should be sufficient for most RPM troubleshooting situations.
debug aaa authorization	This debug output provides more specific information and shows the actual DNIS numbers passed and call types used.

Successful Resource Pool Connection

The following sample output from the **debug resource-pool** command displays a successful RPM connection. The entries in bold are of particular importance.

```
*Mar  1 02:14:57.439: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:21
*Mar  1 02:14:57.439: RM: event incoming call
```

```
*Mar  1 02:14:57.443: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:21
*Mar  1 02:14:57.447: RM:RPM event incoming call
*Mar  1 02:14:57.459: RPM profile ACME found
*Mar  1 02:14:57.487: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_SUCCESS
 DS0:0:0:0:21
*Mar  1 02:14:57.487: Allocated resource from res_group isdn-ports
*Mar  1 02:14:57.491: RM:RPM profile "ACME", allocated resource "isdn-ports" successfully
*Mar  1 02:14:57.495: RM state:RM_RPM_RES_ALLOCATING event:RM_RPM_RES_ALLOC_SUCCESS
DS0:0:0:0:21
*Mar  1 02:14:57.603: %LINK-3-UPDOWN: Interface Serial0:21, changed state to up
*Mar  1 02:15:00.879: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:21, changed
state to up
```

Dialer Component

The dialer component contains DNIS groups and is responsible for configuration, and maintenance of counters and statistics. The resource-pool component is responsible for the following:

- DNIS number statistics or counters
- Configuring DNIS groups

Resource Group Manager

Resource groups are created, maintained, allocated, freed, and tallied by the resource group manager. The resource group manager is also responsible for service profiles, which are applied to resources at call setup time. The resource group manager is responsible for:

- Allocating resources when the profile has been authorized and a valid resource group is received
- Statistics or configuration of resource groups
- Configuring or applying service profiles to resource groups
- Collecting DNIS number information for channel-associated signaling calls

Signaling Stack

The signaling stacks currently supported in resource pooling are CAS and ISDN. The signaling stack delivers the incoming call to the resource-pool dispatcher and provides call-type and DNIS number information to the resource-pool dispatcher. Depending on configuration, call connect attempts may fail if the signaling stacks do not send the DNIS number and the call type to the resource-pool dispatcher. Call attempts will also fail if signaling stacks disconnect prematurely, not giving enough time for authorization or resource allocation processes to complete.

Therefore, investigate the signaling stack when call attempts or call treatment behavior does not meet expectations. For ISDN, the **debug isdn q931** command can be used to isolate errors between resource pooling, signaling stack, and switch. For CAS, the **debug modem csm**, **service internal**, and **modem-mgmt csm debug-rbs** commands are used on Cisco AS5200 and Cisco AS5300 access servers, while the **debug csm** and **debug trunk cas port** *number* **timeslots** *number* commands are used on the Cisco AS5800 access server.

AAA Component

In context with resource pooling, the AAA component is responsible for the following:

- Authorization of profiles between the resource-pool dispatcher and local or external resource-pool manager
- Accounting messages between the resource-pool dispatcher and external resource-pool manager for resource allocation
- VPDN authorization between VPDN and the local or external resource-pool manager
- VPDN accounting messages between VPDN and the external resource-pool manager
- Overflow accounting records between the AAA server and resource-pool dispatcher
- Resource connect speed accounting records between the AAA server and resource group

VPDN Component

The VPDN component is responsible for the following:

- Creating VPDN groups and profiles
- Searching or matching groups based on domain or DNIS
- Maintaining counts and statistics for the groups and profiles
- Setting up the tunnel between the NAS/LAC and HGW/LNS

The VPDN component interfaces with AAA to get VPDN tunnel authorization on the local or remote resource-pool manager. VPDN and AAA debugging traces should be used for troubleshooting.

Troubleshooting DNIS Group Problems

The following output from the **debug resource-pool** command displays a customer profile that is not found for a particular DNIS group:

```
*Mar  1 00:38:21.011: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:3
*Mar  1 00:38:21.011: RM: event incoming call
*Mar  1 00:38:21.015: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:3
*Mar  1 00:38:21.019: RM:RPM event incoming call
*Mar  1 00:38:21.103: RPM no profile found for call-type digital in default DNIS number
*Mar  1 00:38:21.155: RM:RPM profile rejected do not allocate resource
*Mar  1 00:38:21.155: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:3
*Mar  1 00:38:21.163: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:3
```

Troubleshooting Call Discriminator Problems

The following output from the **debug resource-pool** command displays an incoming call that is matched against a call discriminator profile:

```
*Mar  1 00:35:25.995: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:4
*Mar  1 00:35:25.999: RM: event incoming call
*Mar  1 00:35:25.999: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:4
*Mar  1 00:35:26.003: RM:RPM event incoming call
*Mar  1 00:35:26.135: RM:RPM profile rejected do not allocate resource
*Mar  1 00:35:26.139: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:4
*Mar  1 00:35:26.143: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:4
```

Troubleshooting Customer Profile Counts

The following output from the **debug resource-pool** command displays what happens once the customer profile limits have been reached:

```
*Mar  1 00:43:33.275: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:9
*Mar  1 00:43:33.279: RM: event incoming call
*Mar  1 00:43:33.279: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:9
*Mar  1 00:43:33.283: RM:RPM event incoming call
*Mar  1 00:43:33.295: RPM count exceeded in profile ACME
*Mar  1 00:43:33.315: RM:RPM profile rejected do not allocate resource
*Mar  1 00:43:33.315: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_FAIL DS0:0:0:0:9
*Mar  1 00:43:33.323: RM state:RM_RPM_DISCONNECTING event:RM_RPM_DISC_ACK DS0:0:0:0:9
```

Troubleshooting Resource Group Counts

The following output from the **debug resource-pool** command displays the resources within a resource group all in use:

```
*Mar  1 00:52:34.411: RM state:RM_IDLE event:DIALER_INCALL DS0:0:0:0:19
*Mar  1 00:52:34.411: RM: event incoming call
*Mar  1 00:52:34.415: RM state:RM_DNIS_AUTHOR event:RM_DNIS_RPM_REQUEST DS0:0:0:0:19
*Mar  1 00:52:34.419: RM:RPM event incoming call
*Mar  1 00:52:34.431: RPM profile ACME found
*Mar  1 00:52:34.455: RM state:RM_RPM_RES_AUTHOR event:RM_RPM_RES_AUTHOR_SUCCESS
DS0:0:0:0:19
*Mar  1 00:52:34.459: All resources in res_group isdn-ports are in use
*Mar  1 00:52:34.463: RM state:RM_RPM_RES_ALLOCATING event:RM_RPM_RES_ALLOC_FAIL
DS0:0:0:0:19
*Mar  1 00:52:34.467: RM:RPM failed to allocate resources for "ACME"
```

Troubleshooting VPDN

Troubleshooting problems that might typically occur are as follows:

- Customer profile is not associated with a VPDN profile or VPDN group (the call will be locally terminated in this case. Regular VPDN can still succeed even if RPM/VPDN fails).
- VPDN profile limits have been reached (call answered but disconnected).
- VPDN group limits have been reached (call answered but disconnected).
- VPDN endpoint is not reachable (call answered but disconnected).

Troubleshooting RPM/VPDN Connection

The following sample output from the **debug resource-pool** command displays a successful RPM/VPDN connection. The entries in bold are of particular importance.

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
*Mar  1 00:15:53.639: Se0:10 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 00:15:53.655: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 6/0/0/0
*Mar  1 00:15:53.659: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
```

```
*Mar  1 00:15:53.695: Se0:10 RM/VPDN: Session has been authorized using
dnis:ACME_dnis_numbers
*Mar  1 00:15:53.695: Se0:10 RM/VPDN/session-reply: NAS name HQ-NAS
*Mar  1 00:15:53.699: Se0:10 RM/VPDN/session-reply: Endpoint addresses 172.16.1.9
*Mar  1 00:15:53.703: Se0:10 RM/VPDN/session-reply: VPDN tunnel protocol l2f
*Mar  1 00:15:53.703: Se0:10 RM/VPDN/session-reply: VPDN Group outgoing-2
*Mar  1 00:15:53.707: Se0:10 RM/VPDN/session-reply: VPDN domain dnis:ACME_dnis_numbers
*Mar  1 00:15:53.767: RM/VPDN: MLP Bundle SOHO Session Connect with 1 Endpoints:
*Mar  1 00:15:53.771:   IP 172.16.1.9 OK
*Mar  1 00:15:53.771: RM/VPDN/rm-session-connect/ACME_VPDN: VP
LIMIT/ACTIVE/RESERVED/OVERFLOW are now 6/1/0/0
*Mar  1 00:15:54.815: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:10, changed
state to up
*Mar  1 00:15:57.399: %ISDN-6-CONNECT: Interface Serial0:10 is now connected to SOHO
```

Troubleshooting Customer/VPDN Profile

The following sample output from the **debug resource-pool** command displays when there is no VPDN group associated with an incoming DNIS group. However, the output from the **debug resource-pool** command, as shown here, does not effectively reflect the problem:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
*Mar  1 03:40:16.483: Se0:15 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 03:40:16.515: Se0:15 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 03:40:16.527: %VPDN-6-AUTHORERR: L2F NAS HQ-NAS cannot locate a AAA server for
Se0:15 user SOHO
*Mar  1 03:40:16.579: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
*Mar  1 03:40:17.539: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:15, changed
state to up
*Mar  1 03:40:17.615: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
*Mar  1 03:40:19.483: %ISDN-6-CONNECT: Interface Serial0:15 is now connected to SOHO
```

Whenever the **debug resource-pool** command offers no further assistance besides the indication that authorization has failed, enter the **debug aaa authorization** command to further troubleshoot the problem. In this case, the **debug aaa authorization** command output appears as follows:

```
*Mar  1 04:03:49.846: Se0:19 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 04:03:49.854: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): Port='DS0:0:0:0:19'
list='default' service=RM
*Mar  1 04:03:49.858: AAA/AUTHOR/RM vpdn-session: Se0:19 (3912941997) user='301001'
*Mar  1 04:03:49.862: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
service=resource-management
*Mar  1 04:03:49.866: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
protocol=vpdn-session
*Mar  1 04:03:49.866: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-protocol-version=1.0
*Mar  1 04:03:49.870: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-nas-state=3278356
*Mar  1 04:03:49.874: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
rm-call-handle=27
*Mar  1 04:03:49.878: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): send AV
multilink-id=SOHO
*Mar  1 04:03:49.878: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): found list "default"
*Mar  1 04:03:49.882: Se0:19 AAA/AUTHOR/RM vpdn-session (3912941997): Method=LOCAL
*Mar  1 04:03:49.886: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
service=resource-management
```

```
*Mar  1 04:03:49.890: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
protocol=vpdn-session
*Mar  1 04:03:49.890: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-protocol-version=1.0
*Mar  1 04:03:49.894: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-nas-state=3278356
*Mar  1 04:03:49.898: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
rm-call-handle=27
*Mar  1 04:03:49.902: Se0:19 AAA/AUTHOR/RM/local (3912941997): Received AV
multilink-id=SOHO
*Mar  1 04:03:49.906: Se0:19 AAA/AUTHOR/VPDN/RM/LOCAL: Customer ACME has no VPDN  group
for session dnis:ACME_dnis_numbers
*Mar  1 04:03:49.922: Se0:19 AAA/AUTHOR (3912941997): Post authorization status = FAIL
```

Troubleshooting VPDN Profile Limits

The following output from the **debug resource-pool** command displays that VPDN profile limits have been reached:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
*Mar  1 04:57:53.762: Se0:13 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 04:57:53.774: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 0/0/0/0
*Mar  1 04:57:53.778: RM/VPDN/ACME_VPDN: Session outgoing-2 rejected due to Session Limit
*Mar  1 04:57:53.798: Se0:13 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 04:57:53.802: %VPDN-6-AUTHORFAIL: L2F NAS HQ-NAS, AAA authorization failure for
Se0:13 user SOHO; At Session Max
*Mar  1 04:57:53.866: %ISDN-6-DISCONNECT: Interface Serial0:13  disconnected from  SOHO,
call lasted 2 seconds
*Mar  1 04:57:54.014: %LINK-3-UPDOWN: Interface Serial0:13, changed state to down
*Mar  1 04:57:54.050: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON DS0:0:0:0:13
*Mar  1 04:57:54.054: RM:RPM event call drop
*Mar  1 04:57:54.054: Deallocated resource from res_group isdn-ports
```

Troubleshooting VPDN Group Limits

The following **debug resource-pool** command display shows that VPDN group limits have been reached. From this display, the problem is not obvious. To troubleshoot further, use the **debug aaa authorization** command described in the "Troubleshooting RPMS" section later in this chapter:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
*Mar  1 05:02:22.314: Se0:17 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 05:02:22.334: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 5/0/0/0
*Mar  1 05:02:22.334: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 05:02:22.358: Se0:17 RM/VPDN/rm-session-request: Authorization failed
*Mar  1 05:02:22.362: %VPDN-6-AUTHORFAIL: L2F NAS HQ-NAS, AAA authorization failure for
Se0:17 user SOHO; At Multilink Bundle Limit
*Mar  1 05:02:22.374: %ISDN-6-DISCONNECT: Interface Serial0:17  disconnected from  SOHO,
call lasted 2 seconds
*Mar  1 05:02:22.534: %LINK-3-UPDOWN: Interface Serial0:17, changed state to down
*Mar  1 05:02:22.570: RM state:RM_RPM_RES_ALLOCATED event:DIALER_DISCON DS0:0:0:0:17
*Mar  1 05:02:22.574: RM:RPM event call drop
*Mar  1 05:02:22.574: Deallocated resource from res_group isdn-ports
```

Troubleshooting VPDN Endpoint Problems

The following output from the **debug resource-pool** command displays that the IP endpoint for the VPDN group is not reachable:

Note Effective with Cisco Release 12.4(11)T, the **L2F protocol** was removed in Cisco IOS software.

```
*Mar  1 05:12:22.330: Se0:21 RM/VPDN/rm-session-request: Allocated vpdn info for domain
NULL MLP Bundle SOHO
*Mar  1 05:12:22.346: RM/VPDN/ACME_VPDN: VP LIMIT/ACTIVE/RESERVED/OVERFLOW are now 5/0/0/0
*Mar  1 05:12:22.350: RM/VPDN/ACME_VPDN: Session reserved for outgoing-2
*Mar  1 05:12:22.382: Se0:21 RM/VPDN: Session has been authorized using
dnis:ACME_dnis_numbers
*Mar  1 05:12:22.386: Se0:21 RM/VPDN/session-reply: NAS name HQ-NAS
*Mar  1 05:12:22.386: Se0:21 RM/VPDN/session-reply: Endpoint addresses 172.16.1.99
*Mar  1 05:12:22.390: Se0:21 RM/VPDN/session-reply: VPDN tunnel protocol l2f
*Mar  1 05:12:22.390: Se0:21 RM/VPDN/session-reply: VPDN Group outgoing-2
*Mar  1 05:12:22.394: Se0:21 RM/VPDN/session-reply: VPDN domain dnis:ACME_dnis_numbers
*Mar  1 05:12:25.762: %ISDN-6-CONNECT: Interface Serial0:21 is now connected to SOHO
*Mar  1 05:12:27.562: %VPDN-5-UNREACH: L2F HGW 172.16.1.99 is unreachable
*Mar  1 05:12:27.578: RM/VPDN: MLP Bundle SOHO Session Connect with 1 Endpoints:
*Mar  1 05:12:27.582:   IP 172.16.1.99 Destination unreachable
```

Troubleshooting RPMS

In general, the **debug aaa authorization** command is not used for RPM troubleshooting unless the **debug resource-pool** command display is too vague. The **debug aaa authorization** command is more useful for troubleshooting with RPMS. Following is sample output:

```
Router# debug aaa authorization

AAA Authorization debugging is on

Router# show debug

General OS:
  AAA Authorization debugging is on
Resource Pool:
  resource-pool general debugging is on
```

The following output from the **debug resource-pool** and **debug aaa authorization** commands shows a successful RPM connection:

```
*Mar 1 06:10:35.450: AAA/MEMORY: create_user (0x723D24) user='301001'
ruser=''port='DS0:0:0:0:12' rem_addr='102' authen_type=NONE service=NONE priv=0
*Mar 1 06:10:35.462: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907):
Port='DS0:0:0:0:12' list='default' service=RM
*Mar 1 06:10:35.466: AAA/AUTHOR/RM call-accept: DS0:0:0:0:12 (2784758907) user= '301001'
*Mar 1 06:10:35.470: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
service=resource-management
*Mar 1 06:10:35.470: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
protocol=call-accept
*Mar 1 06:10:35.474: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-protocol-version=1.0
*Mar 1 06:10:35.478: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-nas-state=7513368
*Mar 1 06:10:35.482: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-call-type=speech
```

```
*Mar 1 06:10:35.486: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-request-type=dial-in
*Mar 1 06:10:35.486: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): send AV
rm-link-type=isdn
*Mar  1 06:10:35.490: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): found list
"default"
*Mar 1 06:10:35.494: DS0:0:0:0:12 AAA/AUTHOR/RM call-accept (2784758907): Method=LOCAL
*Mar 1 06:10:35.498: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received DNIS=301001
*Mar 1 06:10:35.498: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received CLID=102
*Mar 1 06:10:35.502: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907):Received
Port=DS0:0:0:0:12
*Mar 1 06:10:35.506: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
service=resource-management
*Mar 1 06:10:35.510: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
protocol=call-accept
*Mar 1 06:10:35.510: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-protocol-version=1.0
*Mar 1 06:10:35.514: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-nas-state=7513368
*Mar 1 06:10:35.518: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-call-type=speech
*Mar 1 06:10:35.522: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-request-type=dial-in
*Mar 1 06:10:35.526: DS0:0:0:0:12 AAA/AUTHOR/RM/local (2784758907): Received AV
rm-link-type=isdn
*Mar 1 06:10:35.542: AAA/AUTHOR (2784758907): Post authorization status = PASS_REPL
*Mar 1 06:10:35.546: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
service=resource-management
*Mar 1 06:10:35.550: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
protocol=call-accept
*Mar 1 06:10:35.554: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-protocol-version=1.0
*Mar 1 06:10:35.558: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-response-code=overflow
*Mar 1 06:10:35.558: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-handle=47
*Mar 1 06:10:35.562: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-count=2
*Mar 1 06:10:35.566: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-cp-name=ACME
*Mar 1 06:10:35.570: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-rg-name#0=MICA-modems
*Mar 1 06:10:35.574: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-rg-service-name#0=gold
*Mar 1 06:10:35.578: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-treatment=busy
*Mar  1 06:10:35.582: DS0:0:0:0:12 AAA/AUTHOR/RM/call-accept (2784758907): Processing AV
rm-call-type=speech
```

Configuration Examples for RPM

The following sections provide RPM configuration examples:

- Standard Configuration for RPM Example
- Customer Profile Configuration for DoVBS Example
- DNIS Discriminator Profile Example
- CLID Discriminator Profile Example
- Direct Remote Services Configuration Example

- VPDN Configuration Example
- VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example

Standard Configuration for RPM Example

The following example demonstrates a basic RPM configuration:

```
resource-pool enable
resource-pool call treatment resource busy
resource-pool call treatment profile no-answer
!
resource-pool group resource isdn-ports
 range limit 46
resource-pool group resource MICA-modems
 range port 1/0 2/23
!
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
!
resource-pool profile customer DEFAULT
 limit base-size 10
 resource MICA-modems speech service silver
 dnis group default

resource-pool profile discriminator deny_DNIS
 call-type digital
 dnis group bye-bye
!
resource-pool profile service gold
 modem min-speed 33200 max-speed 56000 modulation v90
resource-pool profile service silver
 modem min-speed 19200 max-speed 33200 modulation v34
!
resource-pool aaa protocol local
!
dialer dnis group ACME_dnis_numbers
 number 301001
dialer dnis group bye-bye
 number 301005
```

Tip

- Replace the command string **resource isdn-ports digital** in the previous example with **resource isdn-ports speech** to set up DoVBS. See the section, "Customer Profile Configuration for DoVBS Example," for more information.

  Digital calls to 301001 are associated with the customer ACME by using the resource group "isdn-ports."
- Speech calls to 301001 are associated with the customer ACME by using the resource group "mica-modems" and allow for V.90 connections (anything less than V.90 is also allowed).
- Digital calls to 301005 are denied.
- All other speech calls to any other DNIS number are associated with the customer profile "DEFAULT" by using the resource group "mica-modems" and allow for V.34 connections (anything more than V.34 is not allowed; anything less than V.34 is also allowed).

- All other digital calls to any other DNIS number are not associated with a customer profile and are therefore not allowed.
- The customer profile named "DEFAULT" serves as the default customer profile for speech calls only. If the solution uses an external RPMS server, this same configuration can be used for backup resource pooling if communication is lost between the NAS and the RPMS.

Customer Profile Configuration for DoVBS Example

To allow ISDN calls with a speech bearer capability to be directed to digital resources, make the following change (highlighted in bold) to the configuration shown in the previous section, "Standard Configuration for RPM Example":

```
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports speech
 dnis group ACME_dnis_numbers
```

This change causes ISDN speech calls (in addition to ISDN digital calls) to be directed to the resource "isdn-ports"; thus, ISDN speech calls provide DoVBS.

DNIS Discriminator Profile Example

The following is sample configuration for a DNIS discriminator. It shows how to enable resource pool management, configure a customer profile, create DNIS groups, and add numbers to the DNIS groups.

```
aaa new-model
!
! Enable resource pool management
resource-pool enable
!
resource-pool group resource digital
 range limit 20
!
! Configure customer profile
resource-pool profile customer cp1
 limit base-size all
 limit overflow-size 0
 resource digital digital
 dnis group ok
!
!
isdn switch-type primary-5ess
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
interface Loopback1
 ip address 192.168.0.0 255.255.255.0
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
```

```
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
 no peer default ip address
 ppp authentication chap
!
! Configure DNIS groups
dialer dnis group blot
 number 5552003
 number 3456789
 number 2345678
 number 1234567
!
dialer dnis group ok
 number 89898989
 number 5551003
!
dialer-list 1 protocol ip permit
```

CLID Discriminator Profile Example

The following is a sample configuration of a CLID discriminator. It shows how to enable resource pool management, configure resource groups, configure customer profiles, configure CLID groups and DNIS groups, and add them to discriminator profiles.

```
version xx.x
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname cisco-machine
!
aaa new-model
aaa authentication login djm local
!
username eagle password ***
username infiniti password ***
spe 1/0 1/7
 firmware location system:/ucode/mica_port_firmware
spe 2/0 2/7
 firmware location system:/ucode/mica_port_firmware
!
! Enable resource pool management
resource-pool enable
!
! Configure resource groups
resource-pool group resource digital
 range limit 20
!
! Configure customer profiles
resource-pool profile customer cp1
 limit base-size all
 limit overflow-size 0
 resource digital digital
dnis group ok
!
! Configure discriminator profiles
resource-pool profile discriminator baadaabing
 call-type digital
clid group stompIt
```

```
!
resource-pool profile discriminator baadaaboom
 call-type digital
 clid group splat
!
ip subnet-zero
!
isdn switch-type primary-5ess
chat-script dial ABORT BUSY "" AT OK "ATDT \T" TIMEOUT 30 CONNECT \c
!
!
mta receive maximum-recipients 0
partition flash 2 8 8
!
!
controller T1 0
 framing esf
 clock source line primary
 linecode b8zs
 pri-group timeslots 1-24
!
controller T1 1
 shutdown
 clock source line secondary 1
!
controller T1 2
 shutdown
 clock source line secondary 2
!
controller T1 3
 shutdown
 clock source line secondary 3
!
controller T1 4
 shutdown
 clock source line secondary 4
!
controller T1 5
 shutdown
 clock source line secondary 5
!
controller T1 6
 shutdown
 clock source line secondary 6
!
controller T1 7
 shutdown
 clock source line secondary 7
!
interface Loopback0
 ip address 192.168.12.1 255.255.255.0
!
interface Loopback1
 ip address 192.168.15.1 255.255.255.0
!
interface Loopback2
 ip address 192.168.17.1 255.255.255.0
!
interface Ethernet0
 ip address 10.0.39.15 255.255.255.0
 no ip route-cache
 no ip mroute-cache
!
interface Serial0
```

```
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial1
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial2
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial3
 no ip address
 no ip route-cache
 no ip mroute-cache
 shutdown
 no fair-queue
 clockrate 2015232
!
interface Serial0:23
 ip unnumbered Loopback1
 encapsulation ppp
 ip mroute-cache
 dialer-group 1
 isdn switch-type primary-5ess
 no peer default ip address
 ppp authentication chap pap
!
interface FastEthernet0
 ip address 10.0.38.15 255.255.255.0
 no ip route-cache
 no ip mroute-cache
 duplex half
 speed 100
!
!
ip local pool default 192.168.13.181 192.168.13.226
ip classless
ip route 172.25.0.0 255.0.0.0 Ethernet0
ip route 172.19.0.0 255.0.0.0 Ethernet0
no ip http server
!
!
! Configure DNIS groups
dialer dnis group blot
 number 4085551003
 number 5552003
 number 2223333
 number 3456789
 number 2345678
 number 1234567
!
```

```
dialer dnis group ok
 number 89898989
 number 4084442002
 number 4085552002
 number 5551003
!
dialer clid group splat
 number 12321224
!
! Configure CLID groups
dialer clid group zot
 number 2121212121
 number 4085552002
!
dialer clid group snip
 number 1212121212
!
dialer clid group stompIt
 number 4089871234
!
dialer clid group squash
 number 5656456
dialer-list 1 protocol ip permit
!
!
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
line 1 96
 no exec
 exec-timeout 0 0
 autoselect ppp
line aux 0
line vty 0 4
 exec-timeout 0 0
transport input none
!
scheduler interval 1000
end
```

Direct Remote Services Configuration Example

The following example shows a direct remote services configuration:

```
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
 aaa group-configuration tahoe
 source  template acme_direct
!
resource-pool profile customer DEFAULT
 limit base-size 10
 resource MICA-modems speech service silver
 dnis group default
resource-pool profile discriminator deny_DNIS
  call-type digital
  dnis group bye-bye
```

```
!
resource-pool profile service gold
 modem min-speed 33200 max-speed 56000 modulation v90
resource-pool profile service silver
 modem min-speed 19200 max-speed 33200 modulation v34
!
resource-pool aaa protocol local
!
template acme_direct
 peer default ip address pool tahoe
 ppp authentication chap isdn-users
 ppp multilink
!
dialer dnis group ACME_dnis_numbers
 number 301001
dialer dnis group bye-bye
 number 301005
```

VPDN Configuration Example

Adding the following commands to those listed in the section "Standard Configuration for RPM Example" earlier in this chapter allows you to use VPDN by setting up a VPDN profile and a VPDN group:

Note If the limits imposed by the VPDN profile are not required, do not configure the VPDN profile. Replace the **vpdn profile** *ACME_VPDN* command under the customer profile ACME with the **vpdn group** *outgoing-2* command.

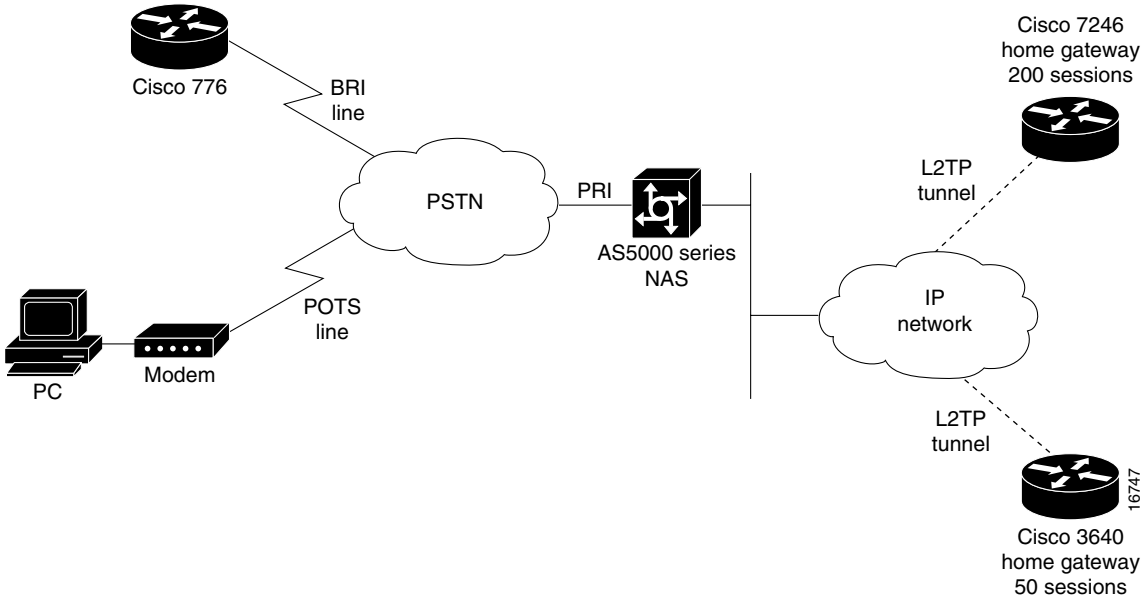
```
resource-pool profile vpdn ACME_VPDN
 limit base-size 6
 limit overflow-size 0
 vpdn group outgoing-2
!
resource-pool profile customer ACME
 limit base-size 30
 limit overflow-size 10
 resource isdn-ports digital
 resource MICA-modems speech service gold
 dnis group ACME_dnis_numbers
!
vpdn profile ACME_VPDN
!
vpdn enable
!
vpdn-group outgoing-2
 request dialin
  protocol l2f
  dnis ACME_dnis_numbers
 local name HQ-NAS
 initiate-to ip 172.16.1.9
 multilink bundle 1
 multilink link 2
!
dialer dnis group ACME_dnis_numbers
 number 301001
```

VPDN Load Sharing and Backing Up Between Multiple HGW/LNSs Example

Cisco IOS software enables you to balance and back up VPDN sessions across multiple tunnel endpoints (HGW/LNS). When a user or session comes into the NAS/LAC, a VPDN load-balancing algorithm is triggered and applied to the call. The call is then passed to an available HGW/LNS. You can modify this function by limiting the number of sessions supported on an HGW/LNS router and limiting the number of MLP bundles and links.

Figure 45 shows an example of one NAS/LAC that directs calls to two HGW/LNS routers by using the L2TP tunneling protocol. Each router has a different number of supported sessions and works at a different speed. The NAS/LAC is counting the number of active simultaneous sessions sent to each HGW/LNS.

Figure 45 Home Gateway Load Sharing and Backup



In a standalone NAS environment (no RPMS server used), the NAS has complete knowledge of the status of tunnel endpoints. Balancing across endpoints is done by a “least-filled tunnel” or a “next-available round robin” approach. In an RPMS-controlled environment, RPMS has the complete knowledge of tunnel endpoints. However, the NAS still has the control over those tunnel endpoints selected by RPMS.

A standalone NAS uses the following default search criteria for load-balancing traffic across multiple endpoints (HGW/LNS):

- Select any idle endpoint—an HGW/LNS with no active sessions.
- Select an active endpoint that currently has a tunnel established with the NAS.
- If all specified load-sharing routers are busy, select the backup HGW. If all endpoints are busy, report that the NAS cannot find an IP address to establish the call.

Note This default search order criteria is independent of the Cisco RPMS application scenario. A standalone NAS uses a different load-sharing algorithm than the Cisco RPMS. This search criteria will change as future enhancements become available.

The following is an example of VPDN load sharing between multiple HGW/LNSs:

```
vpdn enable
!
vpdn-group outgoing-2
 request dialin
  protocol l2tp
  dnis ACME_dnis_numbers
 local name HQ-NAS
 initiate-to ip 172.16.1.9
 loadsharing ip 172.16.1.9 limit 200
 loadsharing ip 172.16.2.17 limit 50
 backup ip 172.16.3.22
```