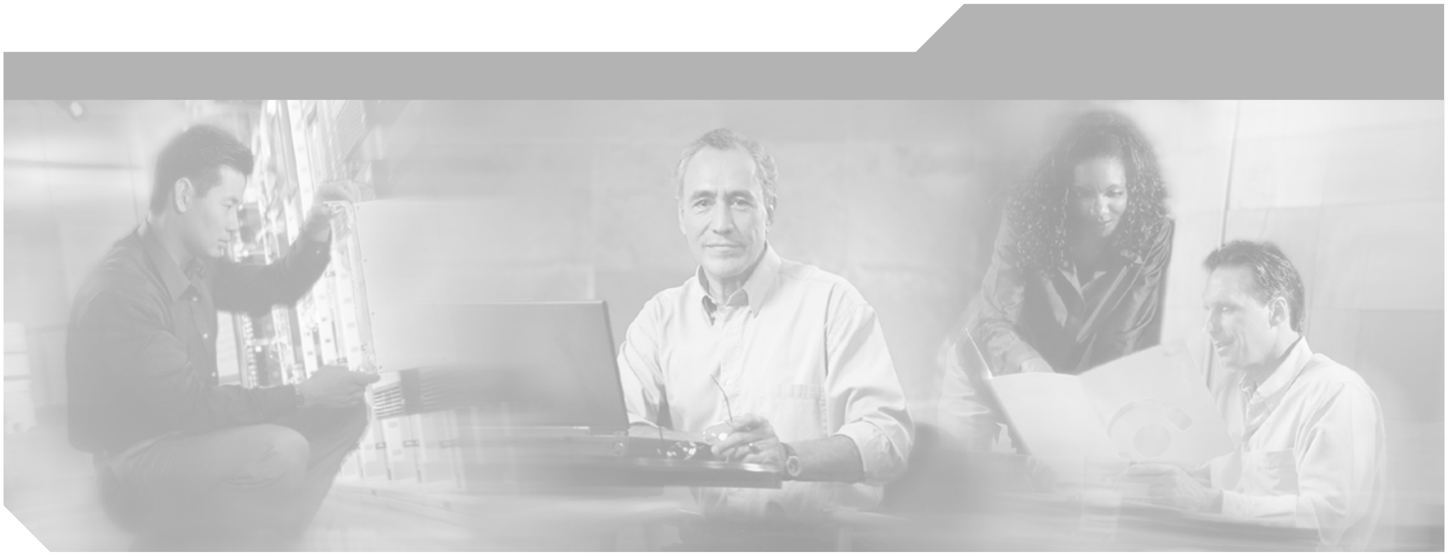


Troubleshooting Cisco Remote Access to MPLS VPN Integration 2.0

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

Customer Order Number:
Text Part Number: OL-2593-02

Troubleshooting Cisco Remote Access to MPLS VPN Integration 2.0

CONTENTS

Preface

Document Overview

Troubleshooting Cisco Remote Access to MPLS VPN Integration 2.0 is designed to provide network administrators with troubleshooting information for the Remote Access to Multiprotocol Label Switching Virtual Private Network Integration solution. Troubleshooting information is provided for each of the remote access methods:

- Dial access
- DSL access
- Cable access

See "Document Organization" for greater detail about these categories.

For quick links to the documentation for the many components of this solution, refer to "Related Documentation".

Intended Audience

It is assumed that administrators of remote access to MPLS VPN integration have experience with installation and acceptance of the products covered by this solution. In addition, it is assumed that the administrator understands the procedures required to upgrade and troubleshoot remote access methods at a basic level.

Typical users of this guide include the follow groups:

- Customers with technical networking background and experience
- Customers who support dial-in users
- System administrators who are familiar with the fundamentals of router-based internetworking, but who may not be familiar with Cisco IOS
- System administrators who are responsible for installing and configuring internetworking equipment, and who are familiar with Cisco IOS

Document Organization

The *Troubleshooting Cisco Remote Access to MPLS VPN Integration 2.0* guide is organized in the following sections:

Chapter 1. Introduction

Describes the general troubleshooting process and gives basic instructions for using debugging commands.

Chapter 2. Troubleshooting Dial Access to MPLS VPN Integration

Provides troubleshooting information for each of the dial access methods and related features.

Chapter 3.Troubleshooting DSL Access to MPLS VPN Integration

Provides troubleshooting information for each of the DSL access methods and related features.

Chapter 4. Troubleshooting Cable Access to MPLS VPN Integration

Provides troubleshooting information for each of the cable access methods and related features.

Related Documentation

The Cisco Remote Access to MPLS VPN Integration 2.0 Documentation Set

In addition to this guide, the Cisco Remote Access to MPLS VPN Integration 2.0 documentation set includes:

- *Cisco Remote Access to MPLS VPN Integration 2.0 Overview and Provisioning Guide*

  http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/rampls2/ovprov/index.htm

  Refer to this document for an overview of remote access to MPLS VPN integration, hardware and software requirements for the various access methods, and procedures for provisioning for each of the access methods and related applications.

  This document also includes a comprehensive list of documentation for all solution hardware and software, with a link to each guide's online location.

- *Cisco Remote Access to MPLS VPN Integration 2.0 Release Notes*

  http://www.cisco.com/univercd/cc/td/doc/product/vpn/solution/rampls2/relnote/index.htm

Related Troubleshooting Documentation

This section lists and provides links to other documents you may find useful for troubleshooting issues related to remote access to MPLS VPN components but outside the scope of this guide.

Troubleshooting Network Access Server Platforms

Troubleshooting Cisco 3600 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:3600

Troubleshooting Cisco AS5300 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:AS5300Troubleshooting Cisco AS5350 routers:

Troubleshooting Cisco AS5400 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:AS5400

Troubleshooting Cisco AS5800 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:AS5800

Troubleshooting Cisco AS5850 series routers:

http://www.cisco.com/univercd/cc/td/doc/product/access/acs_serv/as5850/5850hig/mv_troub.htm

Troubleshooting Virtual Home Gateway Provider Edge Routers

Troubleshooting Cisco 3600 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:3600

Troubleshooting Cisco 6400 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:6400

Troubleshooting Cisco 7200 series routers

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:7200

Troubleshooting Cisco 7500 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:7500

Troubleshooting DSL Equipment

Troubleshooting Cisco 6xx DSL modems:

http://www.cisco.com/univercd/cc/td/doc/product/cable/cab_rout/index.htm

Troubleshooting Cisco 827 routers:

http://www.cisco.com/univercd/cc/td/doc/product/access/acs_fix/827/827swcfg/trble.htm

Troubleshooting Cisco 6015 routers:

http://www.cisco.com/univercd/cc/td/doc/product/dsl_prod/6015/user/hig/index.htm

Troubleshooting Cisco 6130 hardware:

http://www.cisco.com/univercd/cc/td/doc/product/dsl_prod/c6130ni2/userdoc/instgd/04dcch05.htm

Troubleshooting Cisco 6260 routers:

http://www.cisco.com/univercd/cc/td/doc/product/dsl_prod/6260/user/hig2/index.htm

Troubleshooting Cisco 7500 series routers:

http://www.cisco.com/pcgi-bin/Support/PSP/psp_view.pl?p=Hardware:7500

Troubleshooting Cable Equipment

General cable troubleshooting:

http://www.cisco.com/univercd/cc/td/doc/product/cable/cbl_mgt/cbl_trbl/trblsher.htm

Troubleshooting Cisco uBR924 (CPE):

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/ubr924/924ts_5t.htm

Troubleshooting Cisco uBR72xx (PE):

http://www.cisco.com/univercd/cc/td/doc/product/cable/cab_rout/index.htm

Troubleshooting the Core MPLS Network

Troubleshooting MPLS VPN:

http://www.cisco.com/warp/public/105/mpls_vpn_tsh.html.

Obtaining Documentation

These sections explain how to obtain documentation from Cisco Systems.

World Wide Web

You can access the most current Cisco documentation on the World Wide Web at this URL:

http://www.cisco.com

Translated documentation is available at this URL:

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

Documentation CD-ROM

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

Ordering Documentation

You can order Cisco documentation in these ways:

- Registered Cisco.com users (Cisco direct customers) can order Cisco product documentation from the Networking Products MarketPlace:

  http://www.cisco.com/cgi-bin/order/order_root.pl
- Registered Cisco.com users can order the Documentation CD-ROM through the online Subscription Store:

  http://www.cisco.com/go/subscription
- Nonregistered Cisco.com users can order documentation through a local account representative by calling Cisco Systems Corporate Headquarters (California, U.S.A.) at 408 526-7208 or, elsewhere in North America, by calling 800 553-NETS (6387).

Documentation Feedback

You can submit comments electronically on Cisco.com. In the Cisco Documentation home page, click the **Fax** or **Email** option in the “Leave Feedback” section at the bottom of the page.

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

You can submit your comments by mail by using the response card behind the front cover of your document or by writing to the following address:

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

We appreciate your comments.

Obtaining Technical Assistance

Cisco provides Cisco.com as a starting point for all technical assistance. Customers and partners can obtain online documentation, troubleshooting tips, and sample configurations from online tools by using the Cisco Technical Assistance Center (TAC) Web Site. Cisco.com registered users have complete access to the technical support resources on the Cisco TAC Web Site.

Cisco.com

Cisco.com is the foundation of a suite of interactive, networked services that provides immediate, open access to Cisco information, networking solutions, services, programs, and resources at any time, from anywhere in the world.

Cisco.com is a highly integrated Internet application and a powerful, easy-to-use tool that provides a broad range of features and services to help you with these tasks:

- Streamline business processes and improve productivity
- Resolve technical issues with online support
- Download and test software packages
- Order Cisco learning materials and merchandise
- Register for online skill assessment, training, and certification programs

If you want to obtain customized information and service, you can self-register on Cisco.com. To access Cisco.com, go to this URL:

http://www.cisco.com

Technical Assistance Center

The Cisco Technical Assistance Center (TAC) is available to all customers who need technical assistance with a Cisco product, technology, or solution. Two levels of support are available: the Cisco TAC Web Site and the Cisco TAC Escalation Center.

Cisco TAC inquiries are categorized according to the urgency of the issue:

- Priority level 4 (P4)—You need information or assistance concerning Cisco product capabilities, product installation, or basic product configuration.
- Priority level 3 (P3)—Your network performance is degraded. Network functionality is noticeably impaired, but most business operations continue.
- Priority level 2 (P2)—Your production network is severely degraded, affecting significant aspects of business operations. No workaround is available.

- Priority level 1 (P1)—Your production network is down, and a critical impact to business operations will occur if service is not restored quickly. No workaround is available.

The Cisco TAC resource that you choose is based on the priority of the problem and the conditions of service contracts, when applicable.

Cisco TAC Web Site

You can use the Cisco TAC Web Site to resolve P3 and P4 issues yourself, saving both cost and time. The site provides around-the-clock access to online tools, knowledge bases, and software. To access the Cisco TAC Web Site, go to this URL:

http://www.cisco.com/tac

All customers, partners, and resellers who have a valid Cisco service contract have complete access to the technical support resources on the Cisco TAC Web Site. The Cisco TAC Web Site requires a Cisco.com login ID and password. If you have a valid service contract but do not have a login ID or password, go to this URL to register:

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

If you are a Cisco.com registered user, and you cannot resolve your technical issues by using the Cisco TAC Web Site, you can open a case online by using the TAC Case Open tool at this URL:

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

If you have Internet access, we recommend that you open P3 and P4 cases through the Cisco TAC Web Site.

Cisco TAC Escalation Center

The Cisco TAC Escalation Center addresses priority level 1 or priority level 2 issues. These classifications are assigned when severe network degradation significantly impacts business operations. When you contact the TAC Escalation Center with a P1 or P2 problem, a Cisco TAC engineer automatically opens a case.

To obtain a directory of toll-free Cisco TAC telephone numbers for your country, go to this URL:

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

Before calling, please check with your network operations center to determine the level of Cisco support services to which your company is entitled: for example, SMARTnet, SMARTnet Onsite, or Network Supported Accounts (NSA). When you call the center, please have available your service agreement number and your product serial number.

CHAPTER 1

Introduction

This chapter is designed to provide a general introduction to common operations used in troubleshooting Cisco remote access to Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN) integration over a shared infrastructure. For specifics of troubleshooting each remote access method, see the chapters that follow:

- Chapter 2, "Troubleshooting Dial Access to MPLS VPN Integration"
- Chapter 3, "Troubleshooting DSL Access to MPLS VPN Integration"
- Chapter 4, "Troubleshooting Cable Access to MPLS VPN Integration"

For component overviews and technological descriptions of remote access to MPLS VPN, see the Cisco Remote Access to MPLS VPN Integration 2.0 Overview and Provisioning Guide.

General Troubleshooting Process

For all remote access methods, you follow a similar process to troubleshoot problems:

Step 1 Verify that the configuration on each device is correct. For example, in an L2TP dial-in solution using the service provider's AAA RADIUS server for user authorization, authentication, and accounting, you would verify the configuration on the network access server (NAS), virtual home gateway provider edge router (VHG/PE), and the AAA RADIUS server.

Step 2 Identify the main events in the call flow. In the following chapters, you will find a sample topology and call flow for each of the remote access methods.

Step 3 If possible, identify where in the call flow the problem may originate. Check "Potential Problem Areas" to see if your problem is listed, along with suggested troubleshooting topics.

Step 4 Troubleshoot the appropriate events in the call flow using show or debug commands. If you are not sure which events to focus on, proceed from start to finish. In each section, we describe how to troubleshoot each major event in the call flow. Information on initiating and viewing command output is provided in "Initiating and Viewing Command Output".

Initiating and Viewing Command Output

For troubleshooting any remote access method, you typically view and analyze the output of show or debug commands. As you know if you have done configuration and provisioning for your remote access to MPLS VPN integration solution, Cisco IOS software provides the capability to configure Cisco routers and switches using command-line interface (CLI) commands. It also provides the capability to troubleshoot problems in the network by using show or debug commands relevant to various events in the call flow.

When entering commands:

- Use the question mark (?) and arrow keys to help enter commands.
- Use the appropriate command mode (see "Command Modes") for the command you want to enter. Each command mode restricts you to a set of commands.
- Use the forward slash (/) command syntax to identify interface and port locations (*slot/port*). The slot identification number is the first number identified in the command syntax.

Command Modes

When you use the CLI, a command interpreter called EXEC is employed by the operating system to translate any command and execute its operation. This command interpreter has two access modes, user and privileged, which provide security to the respective command levels. Each command mode restricts you to a subset of mode-specific commands.

debug commands are issued in privileged EXEC mode. You enter privileged mode from user mode. Once in privileged mode, you can enter debugging commands.

Table 1-1 shows how to enter and move between user and privileged modes.

Table 1-1   Common Command Modes

Command Mode	Prompt	Access Method	Escape Method
User EXEC	`AS5800>`	Log in.	Use the **exit** or **logout** command to leave the command line interface.
Privileged EXEC	`AS5800#`	From user EXEC mode, enter the **enable** command.	Use the **disable** command to escape back to user EXEC mode. Use the **exit** or **logout** command to leave the command line interface.

Using debug Commands

Before issuing **debug** commands, please refer to "Important Information on Debug Commands" at http://www.cisco.com/warp/customer/793/access_dial/debug.html.

All **debug** commands are entered in privileged EXEC mode, and most debug commands take no arguments. For example, to enable the **debug vpdn events** command, enter the following in privileged EXEC mode at the command line:

```
debug vpdn events
```

If you are using Telnet to connect to the router, enter the following to display the debug output:

```
terminal monitor
```

To turn off the **debug vpdn events** command, in privileged EXEC mode, enter the no form of the command at the command line:

```
no debug vpdn events
```

To display the state of each debugging option, enter the following at the command line in privileged EXEC mode:

```
show debugging
```

Enabling a **debug** command results in output similar to the following example for the **debug vpdn events** command:

```
Router# debug vpdn events
%LINK-3-UPDOWN: Interface Async6, changed state to up
*Mar 2 00:26:05.537: looking for tunnel -- cisco.com --
*Mar 2 00:26:05.545: Async6 VPN Forwarding...
*Mar 2 00:26:05.545: Async6 VPN Bind interface direction=1
*Mar 2 00:26:05.553: Async6 VPN vpn_forward_user bum6@cisco.com is forwarded
%LINEPROTO-5-UPDOWN: Line protocol on Interface Async6, changed state to up
*Mar 2 00:26:06.289: L2F: Chap authentication succeeded for stella.
```

The router continues to generate such output until you enter the corresponding **no debug** command (in this case, **no debug vpdn events**).

Note The output of debug commands can be extensive. To use debug commands effectively, use a focused rather than a scattershot approach. Identify beforehand what you are looking for in a particular debug command and run the minimum number of commands necessary for your objective.

Context-Sensitive Help on Commands

Context-sensitive help is available at any command prompt. Enter a question mark (?) for a list of complete command names, semantics, and command mode command syntax. Use arrow keys at command prompts to scroll through previous mode-specific commands for display.

Note Cycle through mode specific commands at a mode specific prompt.

For a list of available commands, enter a question mark.

```
AS5800> ?
```

To complete a command, enter known characters followed by a question mark (no space).

```
AS5800> s?
```

For a list of command variables, enter the command followed by a space and a question mark.

```
AS5800> show ?
```

Troubleshooting Dial Access to MPLS VPN Integration

Chapter Overview

This chapter is organized in the following sections:

- Troubleshooting dial-in methods of access to Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN):
  - Troubleshooting Dial-in Access, page 2-1
  - Troubleshooting Dial Backup, page 2-31
- Troubleshooting Dial-out Access, page 2-38
  - Troubleshooting L2TP Dial-out Access, page 2-40
  - Troubleshooting Direct ISDN PE Dial-out, page 2-45
- Troubleshooting features that may be used with dial access:
  - Verifying and Troubleshooting Multilink PPP (on the VHG/PE or NAS/PE), page 2-50
  - Verifying and Troubleshooting Multichassis Multilink PPP (MMP), page 2-50
  - Verifying and Troubleshooting On-demand Address Pools, page 2-54
  - Troubleshooting the Framed-Route VRF Aware Feature, page 2-55

Note In the context of L2TP (tunneled) dial methods, the NAS functions as a LAC (L2TP Access Concentrator) and the VHG/PE (virtual home gateway/provider edge router) functions as an LNS (L2TP Network Server). In the call flow diagrams and descriptions, however, we show this simply as "NAS" and "VHG/PE" for consistency with the other methods. "LAC" and "LNS" may appear in debugging output.

Troubleshooting Dial-in Access

This section describes the call flow for each of the two methods of dial-in access, L2TP and direct ISDN PE, and then presents troubleshooting steps that are applicable, with some variations, to both methods. Procedures for verifying correct configuration are given for each method separately.

Dial-in access may include these features:

- Multilink PPP (MLP)—PPP that is split across multiple data links.
- Multichassis MLP (MMP)—MLP with redundant “stacked” NAS/PEs, using a stack group bidding process (SGBP) to manage the allocation of PPP sessions among the members of the stack.
- Address management through overlapping local pools configured on the NAS/PE or overlapping address pools on the SP AAA server, through a DHCP (dynamic host configuration protocol) server, or through on-demand address pools (ODAP).

Understanding Call Flow in L2TP Dial-in Access to MPLS VPN

Figure 2-1 shows an example of the topology for L2TP dial-in access. In this example, MLP is enabled, so that the remote user establishes PPP sessions over two links. Figure 2-2 shows the steps in the corresponding call flow.

Figure 2-1 Topology of L2TP Dial-in Access to MPLS VPN

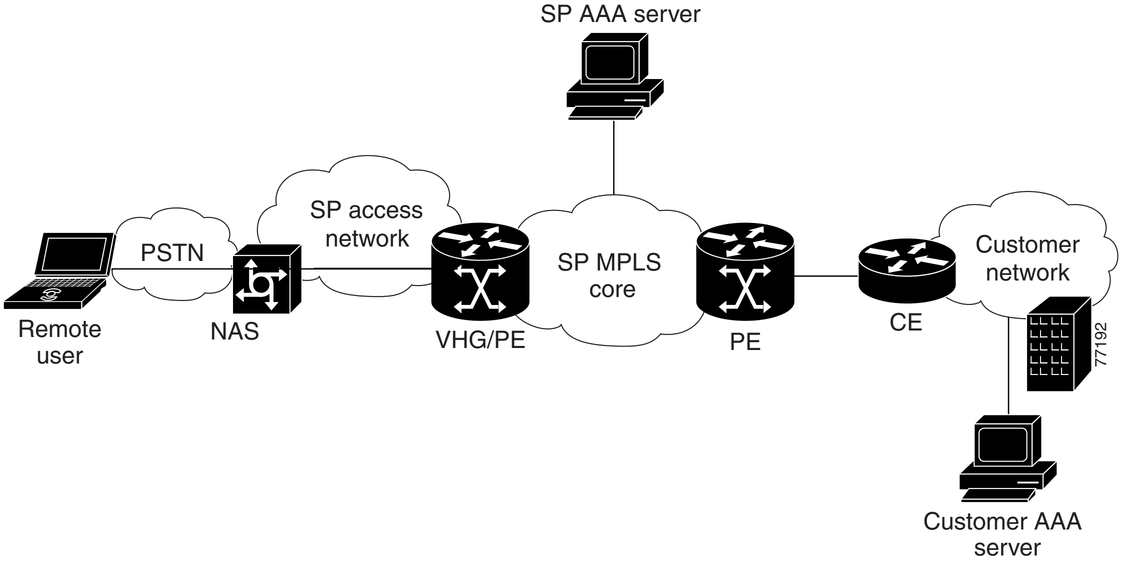


Figure 2-2 Call Flow of L2TP Dial-in Access to MPLS VPN

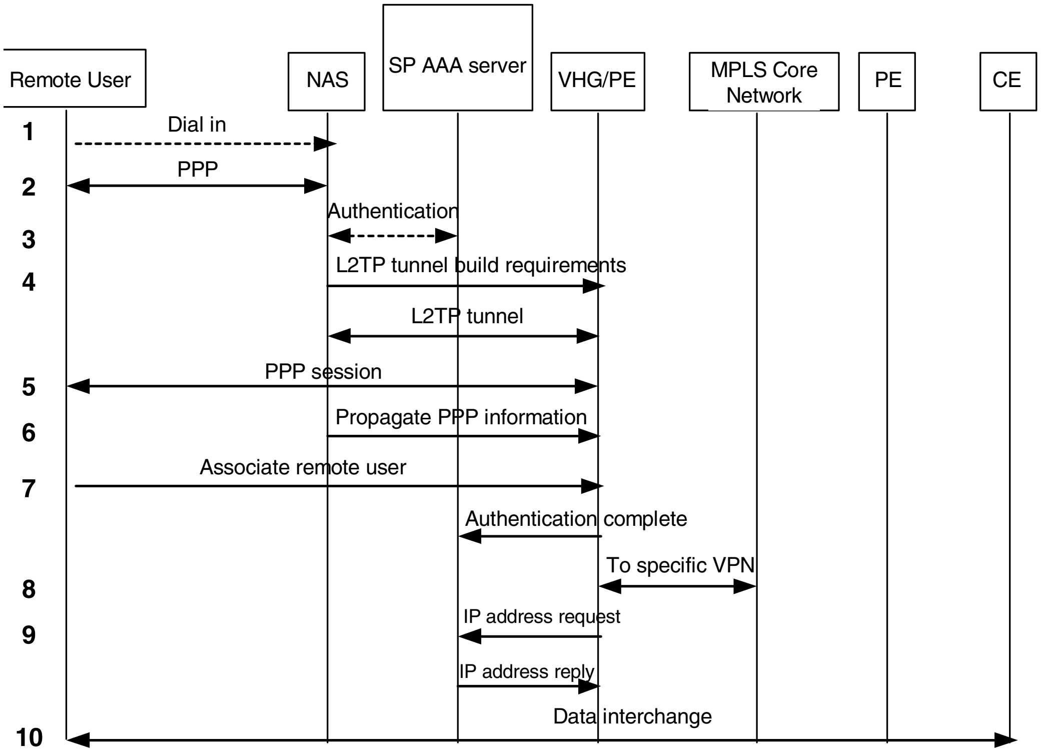


Table 2-1 describes the major dial-in steps shown in Figure 2-2.

Note The call flow assumes that the VPN's VRF (routing table and other information associated with a specific VPN) has already been instantiated on the VHG/PE.

Table 2-1 Troubleshooting Call Flow Events

Step in Call Flow
1. The remote user initiates a PPP connection to the NAS[1] using either analog POTS[2] or ISDN. If MLP is enabled, the session is identified as potentially being part of an MLP bundle.
2. The NAS accepts the connection and a PPP or MLP link is established.
3. The NAS partially authenticates the user with CHAP[3] or PAP[4] and obtains tunnel information. The domain name or DNIS is used to determine whether the user is a VPN client.
4. The NAS initiates a tunnel to the VHG/PE[5] (if an L2TP tunnel does not exist as a result of existing traffic).
5-10(a). The PPP session is created and the connection is extended to terminate on the VHG/PE. The NAS propagates available PPP information.

Step in Call Flow
5-10(b). The VHG/PE completes the authentication, associates the remote user with a specific customer MPLS VPN, and obtains an IP address. The remote user is now part of the customer VPN. Packets can flow to and from the remote user.
If MLP is enabled, the remote user initiates a second PPP (PPP 2 in Figure 2-1) link of the MLP bundle. Above steps are repeated, except that an IP address is not obtained; the existing IP address is used. The remote user can use both PPP sessions. Packets are fragmented across links and defragmented on the VHG/PE, with both MLP bundles put into the same VRF[6].

[1] Network access server

[2] Plain old telephone service

[3] Challenge Handshake Authentication Protocol

[4] Password Authentication Protocol

[5] Virtual home gateway provider edge router

[6] VPN routing/forwarding instance, the routing information for a specific customer VPN site.

Understanding Call Flow in Direct ISDN PE Dial-in Access to MPLS VPN

Direct ISDN PE dial-in access to MPLS VPN is characterized by a NAS functioning as both NAS and PE. In contrast to L2TP dial-in access, the PPP session is placed directly into the appropriate VRF for the MPLS VPN, rather than being forwarded into a network concentrator by a tunneling protocol. Direct dial-in is implemented only with pure ISDN calls, not analog POTS calls.

Figure 2-3 shows an example of the topology for direct dial-in access to MPLS VPN. In this example, MLP is enabled, so that the remote user establishes PPP sessions over two links. Figure 2-4 shows the steps in the corresponding call flow.

Figure 2-3 Topology of Direct Dial-in Access to MPLS VPN

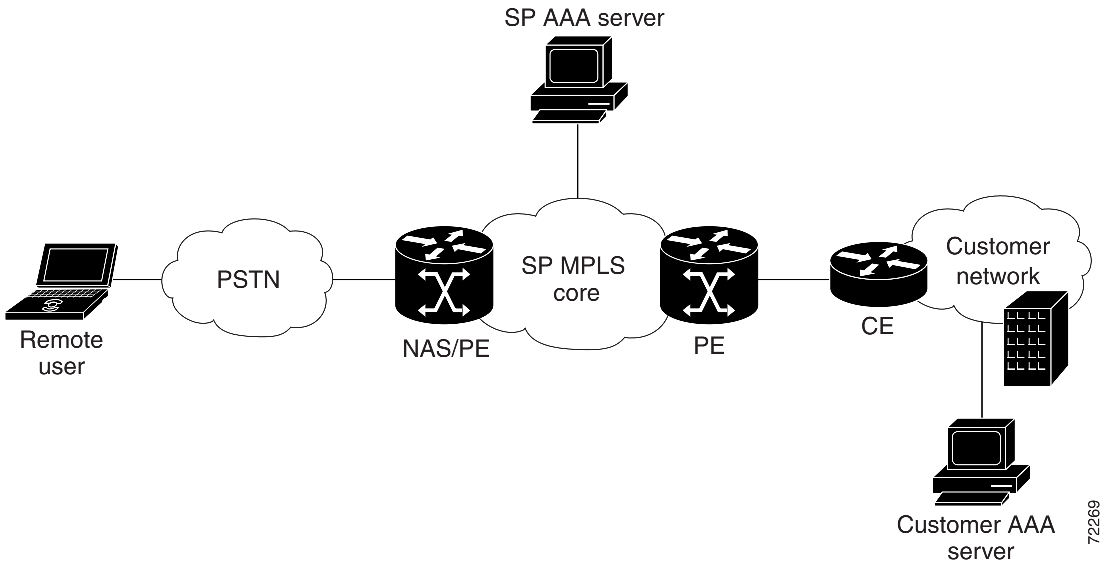
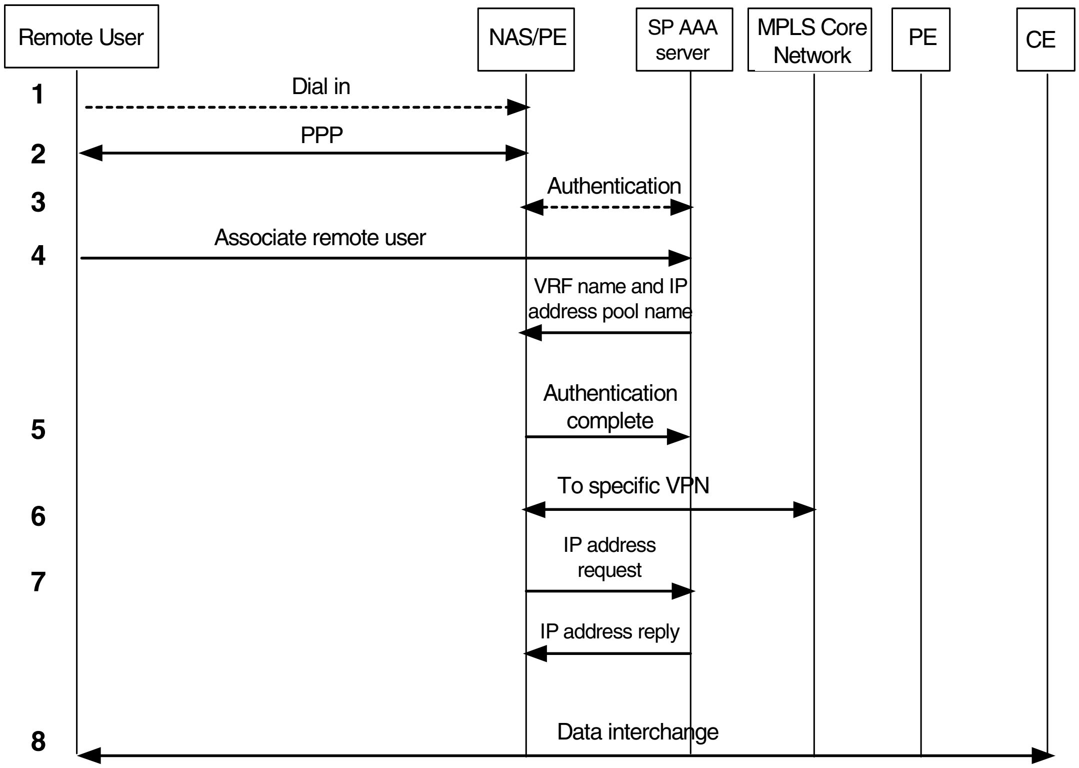


Figure 2-4 Call Flow for Direct Dial-in Access to MPLS VPN



Note The call flow assumes that the VPN's VRF (routing table and other information associated with a specific VPN) has already been instantiated on the NAS/PE.

Table 2-2 Troubleshooting Direct Dial-in Call Flow Events

Step in Call Flow
1. The remote user initiates a PPP connection to the NAS/PE using ISDN. If MLP is enabled, the session is identified as potentially being part of an MLP bundle.
2. The NAS/PE accepts the connection and a PPP or MLP link is established.
3. The NAS/PE partially authenticates the user with CHAP or PAP. The domain name or DNIS is used to determine whether the user is a VPN client.
4. The SP AAA server associates the remote user with a specific VPN and returns the corresponding VRF name to the NAS/PE, along with an IP address pool name.
5 through 8. The NAS/PE completes the authentication, associates the remote user with a specific customer MPLS VPN, and obtains an IP address.

Table 2-2 Troubleshooting Direct Dial-in Call Flow Events (continued)

Step in Call Flow
The remote user is now part of the customer VPN. Packets can flow from/to the remote user.
If MLP is enabled, the remote user initiates a second PPP link of the MLP bundle. The above steps are repeated, except that an IP address is not obtained; the existing IP address is used. The remote user can use both PPP sessions, with packets fragmented across the links and defragmented on the NAS/PE.

Procedure for Troubleshooting Dial-in Access

This section provides a procedure for methodically troubleshooting dial-in remote access problems using a flowchart if/then approach. The steps are summarized in the three related flowcharts below, proceeding from the initial steps in Figure 2-5 to Phase 2 (Figure 2-6) or Phase 3 (Figure 2-7). Details of each numbered step in the flowchart are presented following the figures.

Note If you are troubleshooting direct ISDN PE dial-in, omit steps involving VPDN and L2TP. Steps described for the VHG/PE should be done on the NAS/PE.

Note The following troubleshooting process can be used to resolve common dial-in scenarios in this solution. Exhaustive troubleshooting of all possible combinations of features is beyond the scope of this document.

Figure 2-5 Troubleshooting Dial-in Access, Phase 1

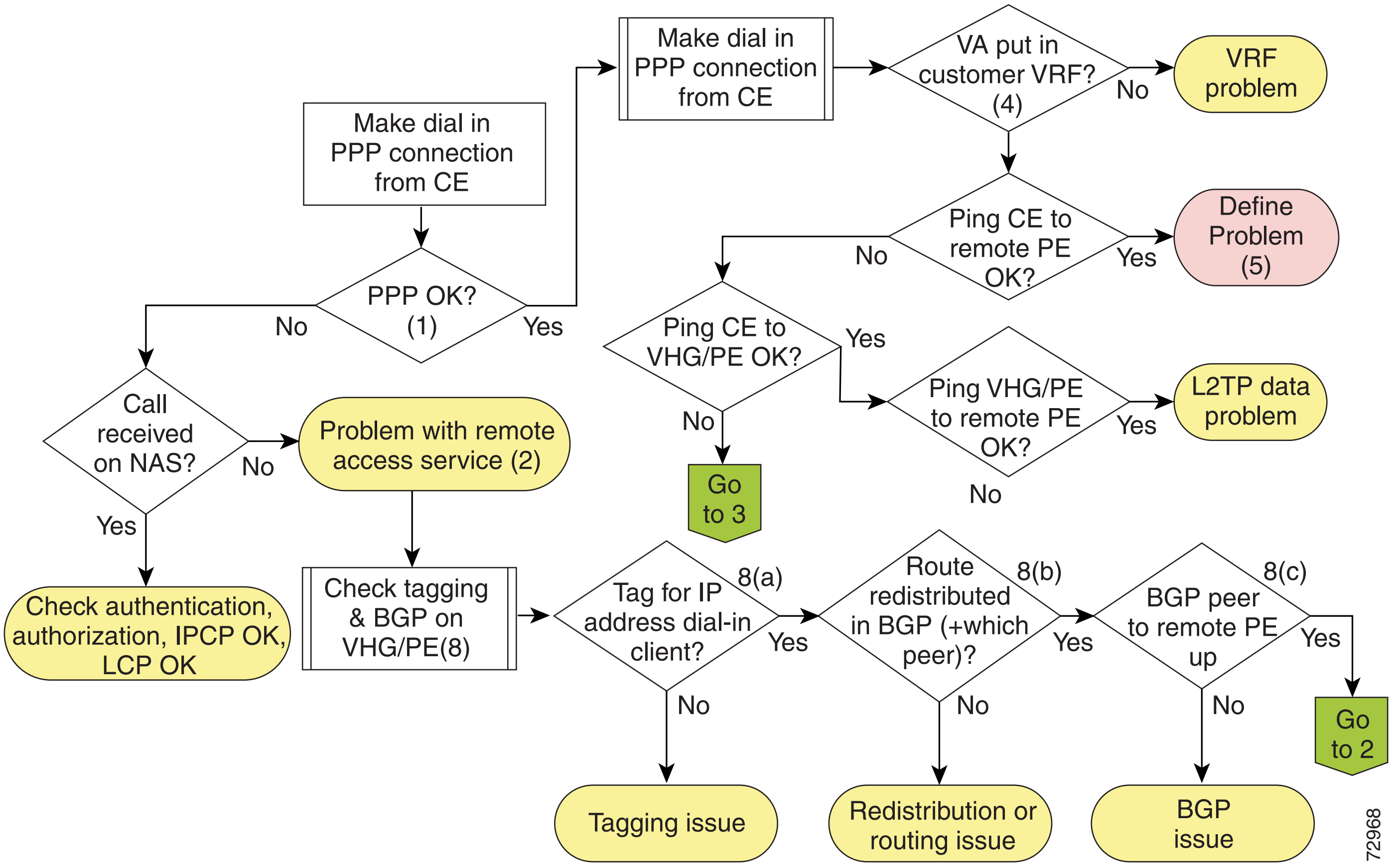

Make dial in PPP connection from CE
PPP OK? (1)
No
Yes
Call received on NAS?
No
Yes
Problem with remote access service (2)
Check authentication, authorization, IPCP OK, LCP OK
Check tagging & BGP on VHG/PE(8)
Make dial in PPP connection from CE
VA put in customer VRF? (4)
No
VRF problem
Ping CE to remote PE OK?
No
Yes
Define Problem (5)
Ping CE to VHG/PE OK?
Yes
No
Go to 3
Ping VHG/PE to remote PE OK?
Yes
No
L2TP data problem
Tag for IP address dial-in client?
8(a)
Yes
No
Tagging issue
Route redistributed in BGP (+which peer)?
8(b)
Yes
No
Redistribution or routing issue
BGP peer to remote PE up
8(c)
Yes
No
BGP issue
Go to 2
72968

Figure 2-6 Troubleshooting Dial-in Access, Phase 2

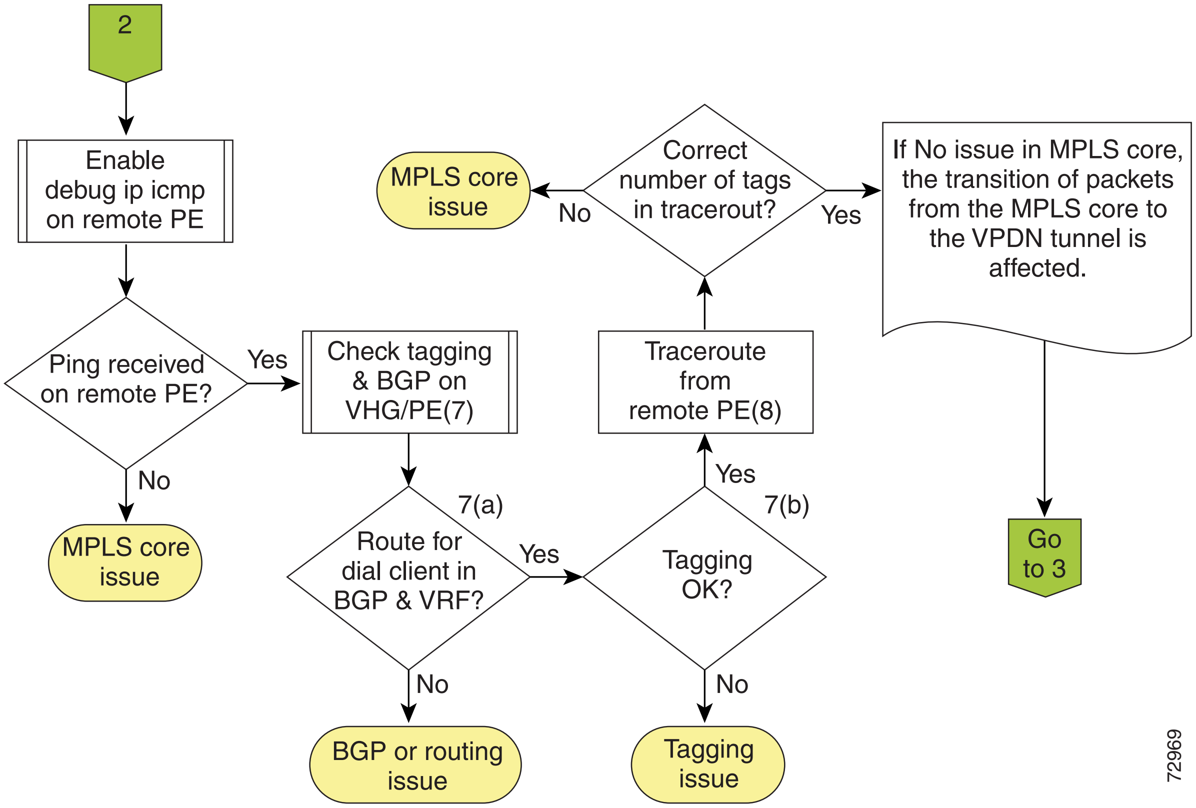
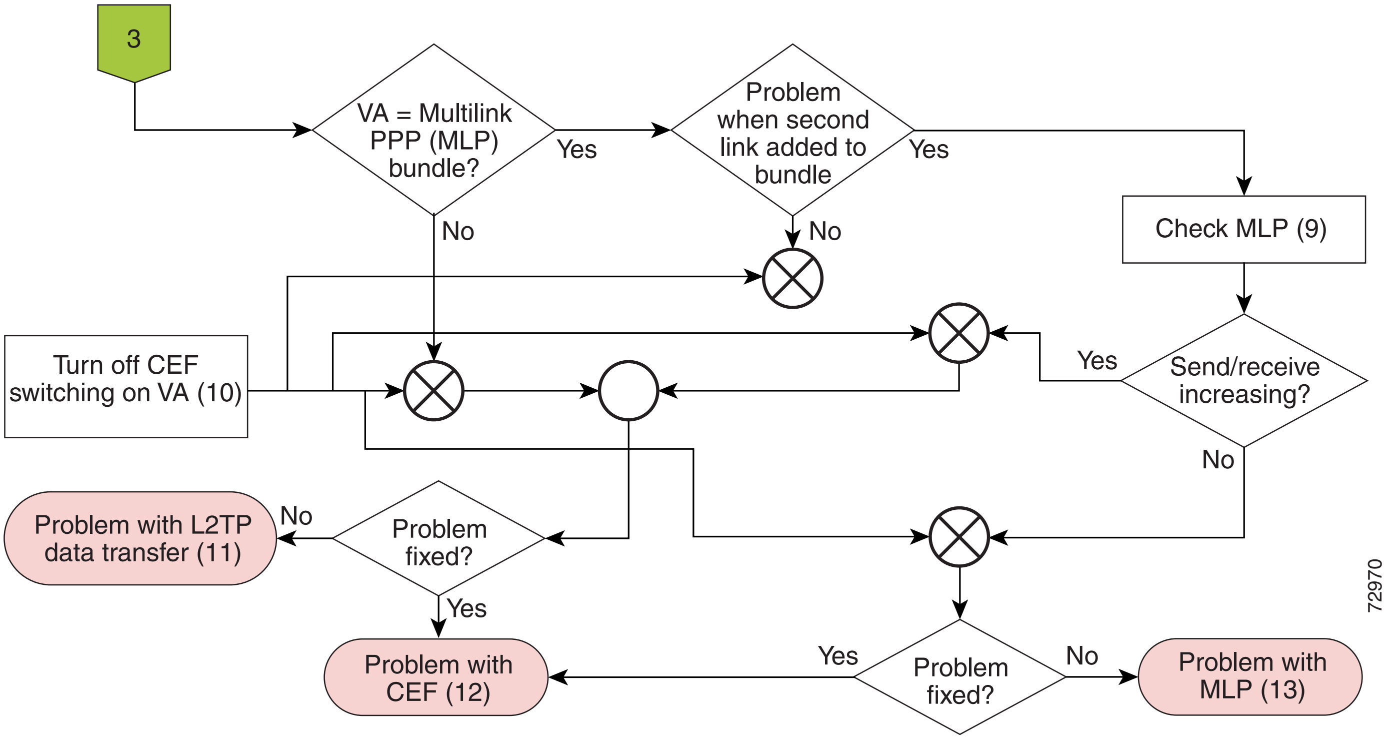

2
Enable debug ip icmp on remote PE
Ping received on remote PE?
Yes
No
MPLS core issue
Check tagging & BGP on VHG/PE(7)
7(a)
Route for dial client in BGP & VRF?
Yes
No
BGP or routing issue
7(b)
Tagging OK?
Yes
No
Tagging issue
Traceroute from remote PE(8)
Correct number of tags in traceroute?
No
MPLS core issue
Yes
If No issue in MPLS core, the transition of packets from the MPLS core to the VPDN tunnel is affected.
Go to 3
72969

Figure 2-7 Troubleshooting Dial-in, Phase 3


3
VA = Multilink PPP (MLP) bundle?
Yes
No
Problem when second link added to bundle
Yes
No
Check MLP (9)
Turn off CEF switching on VA (10)
Yes
Send/receive increasing?
No
Problem with L2TP data transfer (11)
No
Problem fixed?
Yes
Problem with CEF (12)
Yes
Problem fixed?
No
Problem with MLP (13)
72970

Step 1. Is the PPP or MLP Link Established?

On the VHG/PE, use the **show caller** command (Example 2-1) to check the highlighted information. For more detail, use the **show user** command for the user you are interested in (Example 2-2). Use the **debug ppp negotiation** command to check PPP negotiation on either the NAS or the VHG/PE (Example 2-3), depending on where you believe the problem to exist.

For direct ISDN PE dial-in access, use the following ISDN-related commands:

- If the CE appears to be dialing out but is receiving no answer, the NAS/PE may not be connected properly to ISDN. Use the following show commands:
  - **show run**—Use to check that the dialer configuration is correct. If so, check the ISDN connection with **show isdn status** below.
  - **show isdn status**—Use to verify that ISDN is working and there is a connection.
- If ISDN is working correctly, use **debug dialer** to see if packets are triggering the dialer interface.
- If ISDN is not working correctly, use **debug isdn q931** to isolates ISDN information (Layer 2 debugging). Use this only when you have determined that there is an ISDN switch problem. Determine whether it is the CE or user (dial-in) or NAS/PE (dial-out) that is not succeeding in dialing, and run the debug on the device that is having the problem.
- If the CE cannot dial out or does not have a correct connection to ISDN, check the CE device.

Analyzing the Results

IPCP must be negotiated, and the remote end should have an IP address. Both ends must agree on a negotiated IP address. **debug ppp negotiation** should show ACK's received and sent for both IP addresses.

Example 2-1 show caller Output

```
c72d2-3# show caller
                                                                    Active    Idle
  Line      User                      Service      Time      Time
  Vi2       U0001N2P4V1.7@V1.7.com    PPP   L2TP   00:00:02  00:00:03
```

Example 2-2 show user Output

```
c72d2-3# show user U0001N2P4V1.7@V1.7.com
  User: U0001N2P4V1.7@V1.7.com, line Vi2, service PPP L2TP
  PPP: LCP Open, multilink Closed, CHAP (<- none), IPCP
  IP: Local 10.1.7.242, remote 10.1.7.20
  VPDN: NAS lac-lb-V1.7, MID 20, MID Unknown
        HGW c72d2-3, NAS CLID 0, HGW CLID 0, tunnel open
  Counts: 14 packets input, 594 bytes, 0 no buffer
          0 input errors, 0 CRC, 0 frame, 0 overrun
          22 packets output, 806 bytes, 0 underruns
          0 output errors, 0 collisions, 0 interface resets
```

Example 2-3 Sample debug Command Output for Accepting Connection (PPP Connection)

```
[LAC Only]
5300mid# show debug

PPP:
PPP authentication debugging is on
PPP protocol negotiation debugging is on
5300mid#
*Apr  2 12:13:05.413 UTC: As31 LCP: I CONFREQ [Closed] id 10 len 20
*Apr  2 12:13:05.417 UTC: As31 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Apr  2 12:13:05.417 UTC: As31 LCP:    MagicNumber 0x23C67BCA (0x050623C67BCA)
*Apr  2 12:13:05.417 UTC: As31 LCP:    PFC (0x0702)
*Apr  2 12:13:05.417 UTC: As31 LCP:    ACFC (0x0802)
*Apr  2 12:13:05.417 UTC: As31 LCP: Lower layer not up, Fast Starting
*Apr  2 12:13:05.417 UTC: As31 PPP: Treating connection as a dedicated line
*Apr  2 12:13:05.417 UTC: As31 PPP: Phase is ESTABLISHING, Active Open [0sess, 0 load]
*Apr  2 12:13:05.417 UTC: As31 LCP: O CONFREQ [Closed] id 1 len 25
*Apr  2 12:13:05.417 UTC: As31 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Apr  2 12:13:05.417 UTC: As31 LCP:    AuthProto CHAP (0x0305C22305)
*Apr  2 12:13:05.417 UTC: As31 LCP:    MagicNumber 0xF3697B63 (0x0506F3697B63)
*Apr  2 12:13:05.417 UTC: As31 LCP:    PFC (0x0702)
*Apr  2 12:13:05.417 UTC: As31 LCP:    ACFC (0x0802)
*Apr  2 12:13:05.417 UTC: As31 LCP: O CONFACK [REQsent] id 10 len 20
*Apr  2 12:13:05.417 UTC: As31 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Apr  2 12:13:05.417 UTC: As31 LCP:    MagicNumber 0x23C67BCA
(0x050623C67BCA)
*Apr  2 12:13:05.417 UTC: As31 LCP:    PFC (0x0702)
*Apr  2 12:13:05.417 UTC: As31 LCP:    ACFC (0x0802)
*Apr  2 12:13:05.421 UTC: %LINK-3-UPDOWN: Interface Async31, changed state to up
*Apr  2 12:13:05.557 UTC: As31 LCP: I CONFACK [ACKsent] id 1 len 25
*Apr  2 12:13:05.561 UTC: As31 LCP:    ACCM 0x000A0000 (0x0206000A0000)
*Apr  2 12:13:05.561 UTC: As31 LCP:    AuthProto CHAP (0x0305C22305)
*Apr  2 12:13:05.561 UTC: As31 LCP:    MagicNumber 0xF3697B63 (0x0506F3697B63)
```

```
*Apr  2 12:13:05.561 UTC: As31 LCP:    PFC (0x0702)
*Apr  2 12:13:05.561 UTC: As31 LCP:    ACFC (0x0802)
*Apr  2 12:13:05.561 UTC: As31 LCP: State is Open
*Apr  2 12:13:05.561 UTC: As31 PPP: Phase is AUTHENTICATING, by this end [0sess, 0 load]
*Apr  2 12:13:05.561 UTC: As31 CHAP: O CHALLENGE id 1 len 28 from "5300mid"
*Apr  2 12:13:05.701 UTC: As31 CHAP: I RESPONSE id 1 len 39 from "anchan@gcoe.com"
*Apr  2 12:13:05.705 UTC: As31 PPP: Phase is FORWARDING [0 sess, 0 load]
```

Step 2. If the Caller Is Not Seen in (1), Check the Remote Access Service

Use the following debug commands to check common remote access problems, such as VPDN issues, PPP issues, AAA authentication and authorization, RADIUS issues and virtual profile issues:

- **debug vpdn events**—Displays errors and events associated with establishing or terminating L2TP tunnels for VPDNs. Example 2-4 shows sample output.
- **debug vpdn l2x-events**—Displays errors associated with L2X protocol events.
- **debug ppp negotiation—**Enables debugging of the PPP protocol negotiation process.
- **debug ppp authentication**—Enables debugging of errors encountered during remote authentication.
- **debug aaa authentication**—Displays the methods of authentication that are being used and what results these methods produce.
- **debug aaa authorization**—Displays the methods of authorization that are being used and what results these methods produce.

Example 2-5 provides a sample of the debug command output that results if the NAS queries a AAA server for tunnel details.

- **debug aaa per-user—**Displays information about the per-user configuration downloaded from the AAA server.
- **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.
- **debug ip peer—**Displays address activity and contains additional output when pool groups are defined.
- **debug radius**—Displays information associated with the Remote Authentication Dial-in User Server, or RADIUS.

Example 2-6 provides a sample of the debug command output that results when you implement these commands.

For details on troubleshooting L2TP (VPDN) issues, refer to “VPDN Configuration and Troubleshooting”, http://www-tac.cisco.com/Support_Library/Internetworking/VPDN/vpdn_config.0.html.

For details on troubleshooting PPP negotiation, refer to “Dialup Technology: Troubleshooting Techniques”, http://www.cisco.com/warp/public/112/chapter17.htm.

For details on troubleshooting RADIUS AAA, refer to “Diagnosing and Troubleshooting AAA Operations”, http://www.cisco.com/univercd/cc/td/doc/cisintwk/intsolns/secsols/aaasols/c262c6.htm .

Example 2-4 Sample debug Command Output for Establishing L2TP Tunnel

```
[LAC Only]

5300mid# debug vpdn events
VPDN events debugging is on

5300mid# debug vpdn 12x-event                    ^

*Apr  2 13:58:52.137 UTC: %LINK-3-UPDOWN: Interface Async41, changed state to up

*Apr  2 13:58:52.469 UTC: As41 VPDN: Got DNIS string 5551111

*Apr  2 13:58:52.473 UTC: As41 VPDN: Looking for tunnel -- gcoe.com --

*Apr  2 13:58:52.481 UTC: As41 VPDN/RPMS/: Got tunnel info for gcoe.com

*Apr  2 13:58:52.481 UTC: As41 VPDN/RPMS/:   LAC gcoe

*Apr  2 13:58:52.481 UTC: As41 VPDN/RPMS/:   l2tp-busy-disconnect yes

*Apr  2 13:58:52.485 UTC: As41 VPDN/RPMS/:   l2tp-tunnel-password xxxxxx

*Apr  2 13:58:52.485 UTC: As41 VPDN/RPMS/:   IP 10.1.3.1

*Apr  2 13:58:52.485 UTC: As41 VPDN/: curlvl 1 Address 0: 10.1.3.1, priority 1

*Apr  2 13:58:52.485 UTC: As41 VPDN/: Select non-active address 10.1.3.1, priority 1

*Apr  2 13:58:52.485 UTC: Tnl 35054 L2TP: SM State idle

*Apr  2 13:58:52.485 UTC: Tnl 35054 L2TP: O SCCRQ

*Apr  2 13:58:52.485 UTC: Tnl 35054 L2TP: Tunnel state change from idle to wait-ctl-reply

*Apr  2 13:58:52.485 UTC: Tnl 35054 L2TP: SM State wait-ctl-reply

*Apr  2 13:58:52.485 UTC: As41 VPDN: Find LNS process created

*Apr  2 13:58:52.485 UTC: As41 VPDN: Forward to address 10.1.3.1

*Apr  2 13:58:52.485 UTC: As41 VPDN: Pending

*Apr  2 13:58:52.485 UTC: As41 VPDN: Process created

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: I SCCRP from gcoe.com

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: Got a challenge from remote peer, gcoe.com

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: Got a response from remote peer, gcoe.com

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: Tunnel Authentication success

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: Tunnel state change from wait-ctl-reply to
established

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: O SCCCN  to gcoe.com tnlid 46672

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: SM State established

*Apr  2 13:58:52.489 UTC: As41 VPDN: Forwarding...

*Apr  2 13:58:52.489 UTC: As41 VPDN: Bind interface direction=1

*Apr  2 13:58:52.489 UTC: Tnl/Cl 35054/32 L2TP: Session FS enabled

*Apr  2 13:58:52.489 UTC: Tnl/Cl 35054/32 L2TP: Session state change from idle to
wait-for-tunnel

*Apr  2 13:58:52.489 UTC: As41 Tnl/Cl 35054/32 L2TP: Create session

*Apr  2 13:58:52.489 UTC: Tnl 35054 L2TP: SM State established

*Apr  2 13:58:52.489 UTC: As41 Tnl/Cl 35054/32 L2TP: O ICRQ to gcoe.com 46672/0

*Apr  2 13:58:52.493 UTC: As41 Tnl/Cl 35054/32 L2TP: Session state change from
wait-for-tunnel to wait-reply

*Apr  2 13:58:52.493 UTC: As41 VPDN: anchan@gcoe.com is forwarded

*Apr  2 13:58:52.493 UTC: As41 Tnl/Cl 35054/32 L2TP: O ICCN to gcoe.com 46672/45

*Apr  2 13:58:52.493 UTC: As41 Tnl/Cl 35054/32 L2TP: Session state change from wait-reply
to established

*Apr  2 13:58:53.493 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async41,
changed state to up
```

```
[LNS Only]
72vhg2# show debug
VPN:
  L2X protocol events debugging is on
  VPDN events debugging is on
72vhg2#
3d22h: L2TP: I SCCRQ from gcoe tnl 35054
3d22h: Tnl 46672 L2TP: Got a challenge in SCCRQ, gcoe
3d22h: Tnl 46672 L2TP: New tunnel created for remote gcoe, address 10.1.2.183d22h: Tnl
6672 L2TP: O SCCRP  to gcoe tnlid 35054
3d22h: Tnl 46672 L2TP: Tunnel state change from idle to wait-ctl-reply
3d22h: Tnl 46672 L2TP: I SCCCN from gcoe tnl 35054
3d22h: Tnl 46672 L2TP: Got a Challenge Response in SCCCN from gcoe
3d22h: Tnl 46672 L2TP: Tunnel Authentication success
3d22h: Tnl 46672 L2TP: Tunnel state change from wait-ctl-reply to established
3d22h: Tnl 46672 L2TP: SM State established
3d22h: Tnl 46672 L2TP: I ICRQ from gcoe tnl 35054
3d22h: Tnl/Cl 46672/45 L2TP: Session FS enabled
3d22h: Tnl/Cl 46672/45 L2TP: Session state change from idle to wait-connect
3d22h: Tnl/Cl 46672/45 L2TP: New session created
3d22h: Tnl/Cl 46672/45 L2TP: O ICRP to gcoe 35054/32
3d22h: Tnl/Cl 46672/45 L2TP: I ICCN from gcoe tnl 35054, cl 32
3d22h: Tnl/Cl 46672/45 L2TP: Session state change from wait-connect to established
3d22h: Vi3 VPDN: Virtual interface created for anchan@gcoe.com
3d22h: Vi3 VPDN: Set to Async interface
3d22h: Vi3 VPDN: Clone from Vtemplate 25 filterPPP=0 blocking
3d22h: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up
3d22h: Vi3 VPDN: Bind interface direction=2
3d22h: Vi3 VPDN: PPP LCP accepted rcv CONFACK
3d22h: Vi3 VPDN: PPP LCP accepted sent CONFACK
3d22h: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3, changed state to
up
72vhg2#
```

Example 2-5 Sample debug Command Output for Authenticating and Obtaining Tunnel Information

```
[LAC Only]
5300mid# show debug

General OS:
AAA Authentication debugging is on
AAA Authorization debugging is on
Radius protocol debugging is on
5300mid#
*Apr  2 13:25:20.705 UTC: As33 AAA/AUTHOR/FSM: (0): LCP succeeds trivially
*Apr  2 13:25:20.705 UTC: AAA/ACCT/DS0: channel=0, ds1=0, t3=0, slot=0, ds0=0
*Apr  2 13:25:20.709 UTC: %LINK-3-UPDOWN: Interface Async33, changed state to up
*Apr  2 13:25:20.881 UTC: AAA/ACCT/DS0: channel=0, ds1=0, t3=0, slot=0, ds0=0
*Apr  2 13:25:21.041 UTC: AAA: parse name=Async33 idb type=10 tty=33
*Apr  2 13:25:21.041 UTC: AAA: name=Async33 flags=0x11 type=4 shelf=0 slot=0 adapter=0
port=33 channel=0
```

```
*Apr  2 13:25:21.041 UTC: AAA: parse name=Serial0:0 idb type=12 tty=-1
*Apr  2 13:25:21.041 UTC: AAA: name=Serial0:0 flags=0x51 type=1 shelf=0 slot=0 adapter=0
port=0 channel=0
*Apr  2 13:25:21.041 UTC: AAA/ACCT/DS0: channel=0, ds1=0, t3=0, slot=0, ds0=0
*Apr  2 13:25:21.041 UTC: AAA/MEMORY: create_user (0x620DF6C8) user='gcoe.com' ruser=''
port='Async33' rem_addr='async/5551111' authen_type=NONE service=LOGIN priv=0
*Apr  2 13:25:21.041 UTC: Async33 AAA/AUTHOR/VPDN (1549343951): Port='Async33'
list='default' service=NET
*Apr  2 13:25:21.041 UTC: AAA/AUTHOR/VPDN: Async33 (1549343951) user='gcoe.com'
*Apr  2 13:25:21.041 UTC: Async33 AAA/AUTHOR/VPDN (1549343951): send AV service=ppp
*Apr  2 13:25:21.041 UTC: Async33 AAA/AUTHOR/VPDN (1549343951): send AV protocol=vpdn
*Apr  2 13:25:21.041 UTC: Async33 AAA/AUTHOR/VPDN (1549343951): found list "default"
*Apr  2 13:25:21.041 UTC: Async33 AAA/AUTHOR/VPDN (1549343951): Method=radius (radius)
*Apr  2 13:25:21.045 UTC: RADIUS: authenticating to get author data
*Apr  2 13:25:21.045 UTC: RADIUS: ustruct sharecount=2
*Apr  2 13:25:21.045 UTC: RADIUS: Initial Transmit Async33 id 24172.29.51.235:1645,
Access-Request, len 84
*Apr  2 13:25:21.045 UTC:         Attribute 4 6 81010105
*Apr  2 13:25:21.045 UTC:         Attribute 5 6 00000021
*Apr  2 13:25:21.045 UTC:         Attribute 61 6 00000000
*Apr  2 13:25:21.045 UTC:         Attribute 1 13 6A756E69
*Apr  2 13:25:21.045 UTC:         Attribute 30 9 35353531
*Apr  2 13:25:21.045 UTC:         Attribute 2 18 2F3B1919
*Apr  2 13:25:21.045 UTC:         Attribute 6 6 00000005
*Apr  2 13:25:21.053 UTC: RADIUS: Received from id 24 172.29.51.235:1645, Access-Accept,
len 160
*Apr  2 13:25:21.053 UTC:         Attribute 6 6 00000005
*Apr  2 13:25:21.053 UTC:         Attribute 26 30 0000000901187670
*Apr  2 13:25:21.053 UTC:         Attribute 26 29 0000000901177670
*Apr  2 13:25:21.053 UTC:         Attribute 26 35 00000009011D7670
*Apr  2 13:25:21.053 UTC:         Attribute 26 40 0000000901227670
*Apr  2 13:25:21.053 UTC: RADIUS: saved authorization data for user 620DF6C8 at 62136E50
*Apr  2 13:25:21.053 UTC: RADIUS: cisco AVPair "vpdn:tunnel-id=gcoe"
*Apr  2 13:25:21.057 UTC: RADIUS: cisco AVPair "vpdn:tunnel-type=l2tp"
*Apr  2 13:25:21.057 UTC: RADIUS: cisco AVPair "vpdn:ip-addresses=10.1.3.1"
*Apr  2 13:25:21.057 UTC: RADIUS: cisco AVPair "vpdn:l2tp-tunnel-password=bodega"
*Apr  2 13:25:21.057 UTC: AAA/AUTHOR (1549343951): Post authorization status =
PASS_ADD*Apr  2 13:25:21.057 UTC: AAA/AUTHOR/VPDN: Processing AV service=ppp
*Apr  2 13:25:21.057 UTC: AAA/AUTHOR/VPDN: Processing AV protocol=vpdn
*Apr  2 13:25:21.057 UTC: AAA/AUTHOR/VPDN: Processing AV tunnel-id=gcoe
*Apr  2 13:25:21.057 UTC: AAA/AUTHOR/VPDN: Processing AV tunnel-type=l2tp
*Apr  2 13:25:21.057 UTC: AAA/AUTHOR/VPDN: Processing AV ip-addresses=10.1.3.1
*Apr  2 13:25:21.057 UTC: AAA/AUTHOR/VPDN: Processing AV l2tp-tunnel-password=bodega
*Apr  2 13:25:21.057 UTC: AAA/MEMORY: free_user (0x620DF6C8) user='gcoe.com' ruser=''
port='Async33' rem_addr='async/5551111' authen_type=NONE service=LOGIN priv=0
*Apr  2 13:25:21.061 UTC: AAA: parse name=Async33 idb type=10 tty=33
*Apr  2 13:25:21.061 UTC: AAA: name=Async33 flags=0x11 type=4 shelf=0 slot=0 adapter=0
port=33 channel=0
*Apr  2 13:25:21.061 UTC: AAA: parse name=Serial0:0 idb type=12 tty=-1
*Apr  2 13:25:21.061 UTC: AAA: name=Serial0:0 flags=0x51 type=1 shelf=0 slot=0 adapter=0
port=0 channel=0
*Apr  2 13:25:21.061 UTC: AAA/ACCT/DS0: channel=0, ds1=0, t3=0, slot=0, ds0=0
*Apr  2 13:25:21.061 UTC: AAA/MEMORY: create_user (0x620DF6C8) user='anchan@gcoe.com'
ruser='' port='Async33' rem_addr='async/5551111' authen_type=CHAP service=PPP priv=1
*Apr  2 13:25:22.061 UTC: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async33, changed
state to up
```

Example 2-6 Sample debug Command Output for Completing PPP Authentication

```
00:02:18: Vt25 VTEMPLATE: Unable to create and clone vaccess
00:02:18: VTEMPLATE: No unused vaccess, create new vaccess
00:02:18: Vi1 VTEMPLATE: Set default settings with no ip address, encap ppp
00:02:18: Vi1 VTEMPLATE: Hardware address 0003.e412.f800
00:02:18: Vi1 VTEMPLATE: Has a new cloneblk vtemplate, now it has vtemplate
00:02:18: Vi1 VTEMPLATE: ************* CLONE VACCESS1 *****************
00:02:18: Vi1 VTEMPLATE: Clone from Virtual-Template25
interface Virtual-Access1
default ip address
no ip address
encap ppp
no ip address
ip mroute-cache
ppp authentication chap
end
00:02:18: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
00:02:18: RADIUS: ustruct sharecount=1
00:02:18: RADIUS: Initial Transmit Virtual-Access1 id 0 172.29.51.235:1645,
Access-Request, len 98
00:02:18:         Attribute 4 6 8101010E
00:02:18:         Attribute 5 6 00000001
00:02:18:         Attribute 61 6 00000005
00:02:18:         Attribute 1 20 616E6368
00:02:18:         Attribute 30 9 35353531
00:02:18:         Attribute 3 19 01001183
00:02:18:         Attribute 6 6 00000002
00:02:18:         Attribute 7 6 00000001
00:02:18: RADIUS: Received from id 0 172.29.51.235:1645, Access-Accept, len 221
00:02:18:         Attribute 6 6 00000002
00:02:18:         Attribute 7 6 00000001
00:02:18:         Attribute 26 58 0000000901346C63
00:02:18:         Attribute 26 58 0000000901346C63
00:02:18:         Attribute 26 73 0000000901436C63
00:02:18: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf forwarding gcoe"
00:02:18: RADIUS: cisco AVPair "lcp:interface-config#2 = ip unnumbered Loopback200"
00:02:18: RADIUS: cisco AVPair "lcp:interface-config#3 = peer default ip address pool
gcoe.com"
00:02:18: Vi1 VTEMPLATE: Has a new cloneblk AAA, now it has vtemplate/AAA
00:02:18: Vi1 VTEMPLATE: ************* CLONE VACCESS1 *****************
00:02:18: Vi1 VTEMPLATE: Clone from AAA interface Virtual-Access1 ip vrf forwarding gcoe
ip unnumbered Loopback200 peer default ip address pool gcoe.com
end
00:02:18: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf forwarding gcoe" not
applied for ip
00:02:18: RADIUS: cisco AVPair "lcp:interface-config#2 = ip unnumbered Loopback200" not
applied for ip
00:02:18: RADIUS: cisco AVPair "lcp:interface-config#3 = peer default ip address pool
gcoe.com" not applied for ip
00:02:19: Vi1: Pools to search : gcoe.com
00:02:19: Vi1: Pool gcoe.com returned address = 10.40.1.1
00:02:19: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf forwarding gcoe" not
applied for ip
00:02:19: RADIUS: cisco AVPair "lcp:interface-config#2 = ip unnumbered Loopback200" not
applied for ip
00:02:19: RADIUS: cisco AVPair "lcp:interface-config#3 = peer default ip address pool
gcoe.com" not applied for ip
00:02:19: Vi1 AAA/AUTHOR/PER-USER: Event IP_UP
00:02:19: Vi1 AAA/AUTHOR: IP_UP
00:02:19: Vi1 AAA/PER-USER: processing author params.
00:02:19: Vi1 IPCP: Install route to 10.40.1.1
00:02:19: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
```

Step 3. If the PPP Link Is Established, Check Virtual-Access Settings

Before checking the virtual-access settings, verify that the session is established. On the VHG/PE, use the **show vpdn session** and **show caller ip** commands. An IP address should appear in the **show caller ip** results.

If the PPP link is established, check the virtual access interface settings to make sure that the configuration is correct and CEF switching is set up properly. On the virtual access interface:

- Check the configuration. On the VHG/PE, use the **show interfaces virtual-access # configuration** command to check the configuration on the virtual access interface, as shown in Example 2-7.
- Use the **show cef interface** command to check CEF switching capabilities for the virtual access interface, as shown in Example 2-1.
- If CEF switching is not enabled, check that CEF adjacencies are enabled for the virtual-access interface, as in Example 2-1.

Note For MLP, check the virtual-access interface point to the MLP bundle.

Example 2-7 show interfaces virtual-access configuration Output

```
c72d2-3# show interfaces virtual-access 2 configuration
Virtual-Access2 is an L2TP link interface
interface Virtual-Access2 configuration...
ip vrf forwarding V1.7.com
ip unnumbered Loopback7
```

Example 2-8 show cef interface Output

```
c72d2-3# show cef interface virtual-access 2
Virtual-Access2 is up (if_number 50)
  Corresponding hwidb fast_if_number 50
  Corresponding hwidb firstsw->if_number 50
  Internet address is 0.0.0.0/0
  Unnumbered interface. Using address of Loopback7 (10.1.7.242)
  …
  Interface is marked as point to point interface
  Hardware idb is Virtual-Access2
  Fast switching type 7, interface type 21
  IP CEF switching enabled
  IP Feature Fast switching turbo vector
  IP VPN Feature CEF switching turbo vector
  VPN Forwarding table "V1.7.com"
  …
  IP MTU 1464
```

Example 2-9 show adjacency Output

```
c72d2-3# show adjacency virtual-access 2
Protocol      Interface                 Address
TAG           Virtual-Access2           point2point(3) (incomplete)
IP            Virtual-Access2           point2point(6)
c72d2-3# show adjacency virtual-access 2 detail
Protocol      Interface                 Address
```

```
TAG           Virtual-Access2          point2point(3) (incomplete)
                                       0 packets, 0 bytes
                                       Epoch: 0
IP            Virtual-Access2          point2point(6)
                                       2025 packets, 283500 bytes
                                       4500001C56C80000FF1157CC0A0A6823
                                       0A0A910506A506A5000800000202099E
                                       00280000FF030021
                                       CEF   expires: never
                                             refresh: 00:00:44
                                        Epoch: 0
```

Step 4. If Virtual Access Settings Are OK, Check the Customer VRF

From the **show interfaces virtual-access 2 configuration** command used in Step 3, you can identify the VRF into which the dial-in session was placed. The commands below give additional information on the route distinguisher for the VRF.

If the customer is placed into the correct VRF, go on to Step 5.

If the customer is not placed into the correct VRF, there is probably a VRF problem. If the VRF name for this customer is mistyped in the **AAA lcp:interface-config** command, one of two things may happen:

If the error matches an existing VRF configuration on the VHG/PE, the user is placed in the wrong routing table.

If not, a parsing error appears in the virtual template and the session is terminated.

For more information on troubleshooting VRF, refer to "How To Troubleshoot the MPLS VPN", http://www.cisco.com/warp/public/105/mpls_vpn_tsh.html.

Example 2-10 show ip vrf Output

```
c72d2-3# show ip vrf V1.7.com
  Name      Default RD      Interfaces
  V1.7.com  1:7             Serial4/0
                            Virtual-Access2
                            Loopback7
                            Loopback107
```

Example 2-11 show ip vrf interfaces Output

```
c72d2-3# show ip vrf interfaces V1.7.com
Interface        IP-Address      VRF                   Protocol
Virtual-Access1  10.1.7.242      V1.7.com              up
```

Step 5. Ping from the CE to the Remote PE and VHG/PE

If the customer is placed in the correct VRF, check the following:

- Ping from the CE to the remote PE. If the ping succeeds, go on to Step 6. If the ping is unsuccessful, go on to the next bullet point.
- Ping from the CE to the VHG/PE. If the ping succeeds, go on to the next bullet point. If the ping is unsuccessful, go on to

- Ping from the VHG/PE to the remote PE. If the ping is successful, there may be an L2TP data problem. Refer to “VPDN Configuration and Troubleshooting”, http://www-tac.cisco.com/Support_Library/Internetworking/VPDN/vpdn_config.0.html. If the ping is unsuccessful, go on to Step 6.
- If the ping is not successful, the problem may require assistance from TAC. Before contacting TAC, attempt to gather information to define the problem, such as:
  - Small ping packets
  - Large packets
  - A problem with UDP or TCP packets
- Ping from the CE to the remote PE. If the ping to the remote PE is not successful, go on to Step 6 (Figure 3).

Step 6. Check Tagging and BGP on the VHG/PE

When IPCP is negotiated, a host route is placed in the VRF for the dial-in user. This host route creates an FIB entry in the VRF and the VHG/PE assigns it a tag to use when a remote PE forwards traffic to it.

Troubleshoot to determine:

- Is the host route present in the routing table? What is the tag for the host route? Is the route redistributed into BGP and forwarded via the MP-BGP capabilities on the VHG/PE? (Example 2-12).
- (6a) What is the tag value for the VRF route? You can check this with the **show ip cef** command (Example 2-13).
- If CEF is not enabled on the virtual-access interface, collect the tag value for the VRF route tag command (Example 2-14)?
- (6b) Is the host route present in the BGP VRF table and to what BGP peers (remote PEs) is it sent? (Example 2-15)?
- (6c) Is the BGP peer to remote PE up and are multiprotocol BGP capabilities for this peer enabled (Example 2-16 and Example 2-17)?

If tagging and BGP are not OK on the VHG/PE, go on to Step 7. If they are OK, go on to Step 8.

Example 2-12 show ip route Output

```
c72d2-3# show ip route vrf V1.7.com 10.1.7.20
Routing entry for 10.1.7.20/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via bgp 100
  Advertised by bgp 100 metric 1
  Routing Descriptor Blocks:
  * directly connected, via Virtual-Access2
      Route metric is 0, traffic share count is 1
```

Example 2-13 show ip cef Output

```
c72d2-3# show ip cef vrf V1.7.com 10.1.7.20
10.1.7.20/32, version 14, epoch 0, attached, connected, cached adjacency to
Virtual-Access2
tag information set
    local tag: 27
```

```
  via Virtual-Access2, 0 dependencies
    valid cached adjacency
    tag rewrite with Vi2, point2point, tags imposed: {}
```

Example 2-14 show tag forwarding Output

```
c72d2-3# show tag forwarding vrf V1.7.com
Local                             Outgoing                   Prefix             Bytes tag
                                  Outgoing                   Next Hop
tag     tag or VC   or Tunnel Id      switched   interface
27      Untagged   10.1.7.20/32[V]    812700     Vi2        point2point
```

Example 2-15 show ip bgp Output

```
c72d2-3# show ip bgp vpnv4 vrf V1.7.com 10.1.7.20
BGP routing table entry for 1:7:10.1.7.20/32, version 291
Paths: (1 available, best #1)
  Advertised to non peer-group peers:
  10.10.104.12 10.10.104.31
  Local
    0.0.0.0 from 0.0.0.0 (10.10.104.35)
      Origin incomplete, metric 1, localpref 100, weight 32768, valid, sourced, best
      Extended Community: RT:1:7
```

Example 2-16 show ip bgp summary Output

```
c72d2-3# show ip bgp summary
BGP router identifier 10.10.104.35, local AS number 100
…
Neighbor      V      AS  MsgRcvd  MsgSent   TblVer  InQ   OutQ   Up/Down   State/PfxRcd
10.10.104.12  4      100 2971     2963      2       0    0     2d00h     0
10.10.104.31  4      100 3131     2963      2       0    0     2d00h     1
```

Example 2-17 show ip bgp neighbor Output

```
c72d2-3# show ip bgp neighbor 10.10.104.31
BGP neighbor is 10.10.104.31,  remote AS 100, internal link
  BGP version 4, remote router ID 10.10.104.31
  BGP state = Established, up for 2d00h
  Last read 00:00:55, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
…
```

Step 7. Recheck Tagging and BGP on the Remote PE

If the ping is not received on the remote PE, there is probably a problem in the MPLS core. Refer to "How To Troubleshoot the MPLS VPN", http://www.cisco.com/warp/public/105/mpls_vpn_tsh.html.

If the ping is received on the remote PE, proceed with the following troubleshooting.

- (7a) Use the **show ip route** and **show ip bgp** commands on the remote PE to determine if the host route was learned via BGP and is present in the routing table on the remote PE (Example 1and 2). If not, there is probably a problem with BGP or routing. Refer to "Troubleshooting BGP", http://www.cisco.com/warp/public/459/bgp_trouble_main.html.

- (7b) Use the **show ip cef** command to check that tagging is OK on the remote PE (Example 3). If not, there is probably an MPLS tag-switching problem. Refer to “How To Troubleshoot the MPLS VPN”, http://www.cisco.com/warp/public/105/mpls_vpn_tsh.html or “IP Routing Top Issues”, http://www.cisco.com/warp/public/105/top_issues/iprouting/top_issues_rp.shtml.

If MPLS tag-switching and BGP are OK on the remote PE, go on to Step 8.

Note You do not need to recheck BGP peering commands.

Analyzing the Results

In Example 2-20, note that the second tag is 27, the tag value advertised by the VHG/PE to its BGP peers. If the remote PEs have traffic for VRF V1.7.com IP 10.1.7.20, they should use a tag header with the first tag to reach the VHG/PE and the second to indicate to the VHG/PE which VRF this packet belongs to and to what outgoing interface it should be sent.

Example 2-18 show adjacency Output, Remote PE

```
c26d12-1# show ip route vrf V1.7.com 10.1.7.20
Routing entry for 10.1.7.20/32
  Known via "bgp 100", distance 200, metric 1, type internal
  Last update from 10.10.104.35 00:00:30 ago
  Routing Descriptor Blocks:
  * 10.10.104.35 (Default-IP-Routing-Table), from 10.10.104.35, 00:00:30 ago …
```

Example 2-19 show ip bgp Output, Remote PE

```
c26d12-1# show ip bgp vpnv4 vrf V1.7.com 10.1.7.20
BGP routing table entry for 1:7:10.1.7.20/32, version 5103
Paths: (1 available, best #1)
Flag: 0x410
  Not advertised to any peer
  Local
    10.10.104.35 (metric 14) from 10.10.104.35 (10.10.104.35)
      Origin incomplete, metric 1, localpref 100, valid, internal, best
      Extended Community: RT:1:7
```

Example 2-20 show ip cef Output, Remote PE

```
c26d12-1# show ip cef vrf V1.7.com 10.1.7.20
10.1.7.20/32, version 204, cached adjacency 10.10.103.113
0 packets, 0 bytes
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with Et0/1, 10.10.103.113, tags imposed: {57 27}
  via 10.10.104.35, 0 dependencies, recursive
    next hop 10.10.103.113, Ethernet0/1 via 10.10.104.35/32
    valid cached adjacency
    tag rewrite with Et0/1, 10.10.103.113, tags imposed: {57 27}
```

Step 8. Trace the Route from the Remote PE.

If MPLS tag-switching and BGP are OK on the remote PE, use the **traceroute** command to check the route to the VHG/PE or the CE.

Analyzing the Results

The **traceroute** output should show label 27 with either two tags or one tag.

- If there are two tags, then the P router has been traversed but the penultimate hop of the P router immediately before the VHG/PE is not being reached. There may be a problem in the MPLS core. Refer to “How To Troubleshoot the MPLS VPN”, http://www.cisco.com/warp/public/105/mpls_vpn_tsh.html.
- If there is one tag, then the penultimate hop is being reached.
- If there are the correct number of tags, there may be a problem in the transition of packets from the MPLS core to the VPDN tunnel. Go to Step 9 (Figure 3).

Note The **traceroute** command can also be used from the VHG/PE to check if the MPLS tag switching path from VHG/PE to remote PE is OK.

Example 2-21 show traceroute Output

```
c26d12-1# traceroute vrf V1.7.com 10.1.7.20

Type escape sequence to abort.
Tracing the route to 10.1.7.20

  1 10.10.103.113 [MPLS: Labels 57/27 Exp 0] 60 msec 52 msec 52 msec
  2 10.10.103.134 [MPLS: Labels 107/27 Exp 0] 48 msec 52 msec 52 msec
  3 10.10.103.130 [MPLS: Labels 18/27 Exp 0] 52 msec 49 msec 76 msec
  4 10.1.7.242 [MPLS: Label 27 Exp 0] 4 msec 0 msec 4 msec
  5 10.1.7.20 24 msec *  208 msec
```

Step 9. Check MLP on the VHG/PE

If there are no problems in the MPLS core and MPLS tag-switching path is OK, but the VHG/PE is not passing the tag-switched traffic to the NAS/PE via L2TP, do the following:

- Identify if this is an MLP user, then do one of the following:
  - If this is not an MLP user, there may be a problem with L2TP data transfer or CEF switching. Go on to Step 10 to rule out CEF switching issues.
  - If this is an MLP user, check to see if the the expected number of links (bundles) are being created. If so, check to see if send/receive packets are increasing correctly. If not, go on to Step 10 to rule out CEF switching problems.
- If send/receive packets are increasing, go on to Step 10 to rule out CEF switching problems. If packets are not increasing, also go on to Step 10.

Note The following procedure can also be used to troubleshoot unsuccessful pings from the remote PE to the VHG/PE.

Example 2-22 show ppp multilink Output

```
c72d2-3# show ppp multilink
Virtual-Access13, bundle name is U0002N2P4V1.7@V1.7.com
  Bundle up for 00:26:04
  Using relaxed lost fragment detection algorithm.
  15 lost fragments, 21 reordered, 0 unassigned
  9 discarded, 2 lost received, 1/255 load
  0x17E88 received sequence, 0x17E58 sent sequence
  Member links: 1 (max not set, min not set)
    lac-lb-V1.7:Virtual-Access2  (10.10.145.5), since 00:26:04, last rcvd seq 017E86, unsequenced
```

Step 10. Turn Off CEF Switching on the Virtual Access Interface

This step is purely a temporary troubleshooting strategy, not a solution. If you suspect a CEF switching problem, turn off CEF switching to see if the call gets through without it. If so, the problem is in CEF switching. Go on to Step 12.

If this does not solve the problem, the next step depends on what you were investigating before turning off CEF switching. Do one of the following:

- If this is not an MLP user, there is probably a problem with L2TP data transfer. Go on to Step 11.
- If this is an MLP user but there was not a problem in adding a second link to the bundle, there is probably a problem with L2TP data transfer. Go on to Step 11.

Note CEF switching can harmlessly be turned off for troubleshooting purposes. It is not required for placing a route in the VRF or creating an MPLS tag value for the host route. When CEF switching is turned off, traffic destined for the dial-in VPN client simply takes a different switching path (fast switching or process switching) on the VHG/PE.

To turn off CEF switching on the virtual-access interface, enter this configuration command on the virtual-template:

no ip route-cache cef

Step 11. Troubleshoot L2TP Data Transfer

(Not applicable to direct ISDN PE dial-in access) If you have narrowed the problem down to an L2TP data transfer issue, use the following command to collect relevant debugging information:

debug vpdn

To check L2TP data transfer on the NAS, use the command **debug vpdn l2x-data**.

Refer to “VPDN Configuration and Troubleshooting”, http://www-tac.cisco.com/Support_Library/Internetworking/VPDN/vpdn_config.0.html, or contact TAC.

Step 12. Troubleshoot CEF Switching

If you have narrowed the problem down to a CEF switching issue, use the following command to collect relevant debugging information:

debug ip cef drops

Refer to "How to Verify CEF Switching", http://www.cisco.com/warp/public/105/cef_whichpath.html.

Step 13. Troubleshoot MLP

If you have narrowed the problem down to an MLP issue, use the following command to collect relevant debugging information:

debug ppp multilink

See the "Verifying and Troubleshooting Multilink PPP (on the VHG/PE or NAS/PE)" section on page 2-50. Also refer to "Configuring and Troubleshooting Multilink PPP", http://www.cisco.com/warp/public/471/top_issues/access/793_multilink.html or contact TAC.

Verifying Correct Configuration for L2TP Dial-in Access to MPLS to VPN

This section provides the following examples of how you can show information for the events outlined in Figure 2-1 on page 2-2:

- show Commands for NAS, page 2-23
- show Commands for VHG/PE, page 2-25
- show Commands for Overlapping Address Pool, page 2-28

For information on verifying MLP configuration, see "Verifying and Troubleshooting Multilink PPP (on the VHG/PE or NAS/PE)".

The L2TP information provided here applies only to dial access to MPLS VPN integration. For more information about the configuration and troubleshooting tasks associated with L2TP, please refer to *Configuring Virtual Private Networks* (for IOS 12.2) at the following URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fdial_c/fnsprt8/dafvpn.htm

show Commands for NAS

To verify the details of L2TP tunnel setup and session on the NAS, use the following show commands in user EXEC mode:

- **show vpdn—**Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network.
- **show vpdn tunnel—**Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status (summary-style format).
- **show vpdn tunnel all—**Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.
- **show vpdn session—**Displays VPN session information including interface, tunnel, username, packets, status, and window statistics.
- **show user—**Displays information about the active lines on the router per a specific user.
- **show caller—**Displays individual users and consumed resources on the NAS, and active call statistics for large pools of connections, and the absolute and idle times for each user.
- **show caller user** *caller name***—**Displays the show caller command information for a particular user.

- **show vpdn history failure—**Displays the content of the failure history table for the user.
- **show resource-pool call—**(RPMS-specific) Displays all active call information for all customer profiles and resource groups.
- **show resource-pool resource—**(RPMS-specific) Displays resource groups configured in the network access server.

Example 2-23 shows the detailed output that results when you implement these show commands.

Example 2-23 Sample show Command Output for NAS

```
c53c2-1# sh vpdn

L2TP Tunnel and Session Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
29916 29227 c72c3-1       est    10.10.110.1     1701  1

LocID RemID TunID Intf       Username               State  Last Chg Fastswitch
23    30    29916 Se0:30     U0001N3P1V1.2@V1.2.com est    00:00:10 enabled

c53c2-1# sh vpdn tunnel

L2TP Tunnel Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
29916 29227 c72c3-1       est    10.10.110.1     1701  1

c53c2-1# sh vpdn tunnel all

L2TP Tunnel Information Total tunnels 1 sessions 1

Tunnel id 29916 is up, remote id is 29227, 1 active sessions
Tunnel state is established, time since change 00:00:22
Remote tunnel name is c72c3-1
Internet Address 10.10.110.1, port 1701
Local tunnel name is c53c2-1-V1.2
Internet Address 10.10.110.3, port 1701
123 packets sent, 124 received
92982 bytes sent, 92662 received
Control Ns 4, Nr 2
Local RWS 800 (default), Remote RWS 800 (max)
Retransmission time 1, max 1 seconds
Unsent queuesize 0, max 0
Resend queuesize 0, max 2
Total resends 0, ZLB ACKs sent 0
Current nosession queue check 0 of 5
Retransmit time distribution: 0 0 0 0 0 0 0 0 0
Sessions disconnected due to lack of resources 0

c53c2-1# sh vpdn session

L2TP Session Information Total tunnels 1 sessions 1

LocID RemID TunID Intf       Username               State  Last Chg Fastswitch
23    30    29916 Se0:30     U0001N3P1V1.2@V1.2.com est    00:00:32 enabled

c53c2-1# sh caller
                                                Active    Idle
  Line         User               Service       Time      Time
  con 0        -                  TTY           19:04:31  18:58:14
  vty 0        -                  VTY           00:21:59  00:00:00
  Se0:30       U0001N3P1V1.2@V1.2.com \
                                  PPP           00:00:38  00:00:00
5300mid# sh user
    Line       User       Host(s)              Idle       Location
*  0 con 0                idle                 00:00:00
16 tty 16    anchan@jun Async interface          -

Interface  User      Mode                     Idle Peer Address

c53c2-1# sh caller user U0001N3P1V1.2@V1.2.com

  User: U0001N3P1V1.2@V1.2.com, line Se0:30, service PPP
        Active time 00:00:48, Idle time 00:00:00
  Timeouts:            Absolute  Idle
      Limits:          -         -
      Disconnect in:   -         -
  PPP: LCP Open, CHAP (<- none)
  Dialer: Connected, inbound
          Type is ISDN, group Se0:15
  VPDN: NAS c53c2-1-V1.2, MID 23, MID Unknown
        HGW , NAS CLID 0, HGW CLID 0, tunnel open
  Counts: 888 packets input, 252631 bytes, 0 no buffer
          0 input errors, 0 CRC, 0 frame, 0 overrun
          810 packets output, 203706 bytes, 0 underruns
          0 output errors, 0 collisions, 79 interface resets

c53c2-1# sh vpdn hist failure

Table size: 20
Number of entries in table: 1
User: U0001N3P1V1.9@V1.9.com, MID = 15
NAS: Information is not applicable
Gateway: Information is not applicable
Log time: 00:45:47, Error repeat count: 13
Failure type: The remote server closed this session
Failure reason: Result 2, Error 6, Disconnect from PPP

c53c2-1# sh resource-pool call

Shelf 0, slot 0, port 0, channel 30, state RM_RPM_RES_ALLOCATED Customer profile vpdnrpms,
resource group C53c2-1-dig

DNIS number 1020
c53c2-1# sh resource-pool resource

List of Resources:
    C53c2-1-dig
```

show Commands for VHG/PE

To verify the details of the L2TP tunnel setup, PPP sessions, virtual access interface configurations, and local address pool assignment on the VHG/PE, use the following show commands in user EXEC mode:

- **show vpdn**—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network.
- **show vpdn tunnel**—Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status (summary-style format).

- **show vpdn tunnel all**—Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.
- **show vpdn session**—Displays VPN session information including interface, tunnel, username, packets, status, and window statistics.
- **show user**—Displays information about the active lines on the router per a specific user.
- **show caller**—Displays individual users and consumed resources on the NAS, active call statistics for large pools of connections, and the absolute and idle times for each user.
- **show caller user** *caller name*—Displays the "show caller" information for a particular user.
- **show interface virtual-access**—Displays detailed information about the virtual access interface.
- **show virtual access configuration**—Displays detailed information about the virtual access interface configuration.
- **show ip route**—Displays the current state of the routing table, including the IP address, the network mask, protocol, and static routes.
- **show ip local pool**—Displays the address pools that have been downloaded to a Cisco network access server.

Example 2-24 shows the detailed output that results from these show commands:

Example 2-24 Sample show Command Output for VHG/PE

```
72vhg2# sh vpdn

L2TP Tunnel and Session Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
46668 13637 gcoe       est    10.1.2.18      1701  1

LocID RemID TunID Intf    Username      State  Last Chg Fastswitch
9     57    46668 Vi2     anchan@gcoe.c est    4d03h    enabled

72vhg2# sh vpdn tunnel

L2TP Tunnel Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
46668 13637 gcoe       est    10.1.2.18      1701  1

72vhg2# sh vpdn tunnel all

L2TP Tunnel Information Total tunnels 1 sessions 1

Tunnel id 46668 is up, remote id is 13637, 1 active sessions
Tunnel state is established, time since change 4d03h
Remote tunnel name is gcoe
Internet Address 10.1.2.18, port 1701
Local tunnel name is gcoe.com
Internet Address 10.1.3.1, port 1701
71791 packets sent, 71783 received, 1723220 bytes sent, 1723598 received
Control Ns 3, Nr 7
Local RWS 3000 (default), Remote RWS 800
Retransmission time 1, max 1 seconds
Unsent queuesize 0, max 0
Resend queuesize 0, max 1
Total resends 0, ZLB ACKs sent 4
Current nosession queue check 0 of 5
Retransmit time distribution: 0 0 0 0 0 0 0 0 0
```

```
Sessions disconnected due to lack of resources 0

72vhg2# sh vpdn session

L2TP Session Information Total tunnels 1 sessions 1

LocID RemID TunID Intf    Username      State  Last Chg Fastswitch
9     57    46668 Vi2     anchan@gcoe.c est    4d03h    enabled

c72c3-1# sh user
    Line       User       Host(s)              Idle       Location
*  0 con 0                idle                 00:00:00

  Interface      User        Mode                      Idle     Peer Address
  Vi1          U0001N3P1V Virtual PPP (L2TP  ) 00:00:00

c72c3-1# sh caller
                                                Active    Idle
  Line           User               Service       Time      Time
  con 0          -                  TTY           00:22:03  00:00:00
  Vi1            U0001N3P1V1.8@V1.8.com \
                                  PPP   L2TP    00:00:48  00:00:00
c72c3-1# sh caller user U0001N3P1V1.8@V1.8.com

  User: U0001N3P1V1.8@V1.8.com, line Vi1, service PPP L2TP
        Active time 00:01:02, Idle time 00:00:00
  Timeouts:            Absolute  Idle
      Limits:          -         -
      Disconnect in:   -         -
  PPP: LCP Open, multilink Closed, CHAP (<- AAA), IPCP
  IP: Local 12.1.8.250, remote 12.1.8.1
  VPDN: NAS c53c2-1, MID 4, MID Unknown
        HGW , NAS CLID 0, HGW CLID 0, tunnel open
  Counts: 137 packets input, 85970 bytes, 0 no buffer
          0 input errors, 0 CRC, 0 frame, 0 overrun
          21 packets output, 356 bytes, 0 underruns
          0 output errors, 0 collisions, 0 interface resets

c72c3-1# sh int virtual-ac 1

Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback8 (12.1.8.250)
  MTU 1500 bytes, BW 64 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 23/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open, multilink Closed
  Open: IPCP
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 00:05:05
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 6000 bits/sec, 5 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     195 packets input, 128226 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
     23 packets output, 388 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions
```

```
c72c3-1# sh int virtual-ac 1 configur

Virtual-Access1 is an L2TP link interface

Building configuration...

interface Virtual-Access1 configuration...
ip vrf forwarding V1.8.com
ip unnumbered Loopback8
ip mtu 1460
ip mroute-cache
no snmp trap link-status

c72c3-1# sh ip route vrf V1.8.com 12.1.8.250

Routing entry for 12.1.8.250/32
  Known via "connected", distance 0, metric 0 (connected, via interface)
  Redistributing via bgp 100
  Advertised by bgp 100 metric 1
  Routing Descriptor Blocks:
  * directly connected, via Loopback8
      Route metric is 0, traffic share count is 1

c72c3-1# sh ip local pool V1.8-pool

 Pool                     Begin           End             Free  In use
 V1.8-pool                12.1.8.1        12.1.8.10          9       1
Available addresses:
   12.1.8.2
   12.1.8.3
   12.1.8.4
   12.1.8.5
   12.1.8.6
   12.1.8.7
   12.1.8.8
   12.1.8.9
   12.1.8.10
Inuse addresses:
   12.1.8.1           Vi1                        U0001N3P1V1.8@V1.8.com
```

show Commands for Overlapping Address Pool

The overlapping address pool feature allows the administrator to configure overlapping IP address pool groups in order to concurrently use the IP addresses in multiple address spaces.

The design of this feature ensures that a pool name is an explicit selection of a group, or address space. There are, however, a few characteristics about this feature that the administrator must know.

With Overlapping Address Pool, you *can*:

- Assign pools in different groups to a single address space.
- Use with applications that cannot handle duplicate address assignments in their routing table.
- Use on all DSL and Dial platforms that support the current IP local address pools.
- On the Dial platforms, PPP over ISDN sessions will try to use overlapping IP address pools with MPLS VPN configuration.

With Overlapping Address Pool, you *cannot*:

- Associate a pool name with more than one group.
- Associate the pool name default with a named group. It can be used only in the BASE group.

Note The overlapping address feature is independent of the type of configuration used on the dialer.

To verify that an IP address is active, use the **show local pool** command in user EXEC mode. The output shown in Example 2-25 tells you which IP address has been received, and that the call is active and in use.

Example 2-25 Sample show Command Output for Overlapping Address Pool

```
c72c3-1# sh ip local pool V1.28-pool

 Pool                     Begin           End             Free  In use
 ** pool <V1.28-pool> is in group <V1.28-group>
 V1.28-pool               10.1.28.1       10.1.28.10         9       1
Available addresses:
   10.1.28.5
   10.1.28.6
   10.1.28.7
   10.1.28.8
   10.1.28.9
   10.1.28.10
   10.1.28.1
   10.1.28.2
   10.1.28.3
Inuse addresses:
   10.1.28.4          Vi9                         U0001N3P1@V1.28.com
```

Verifying Correct Configuration for Direct ISDN PE Dial-in Access to MPLS VPN

This section provides the following examples of how you can show information for the events outlined in Figure 2-1 on page 2-2:

- show Commands for NAS/PE, page 2-29
- show Commands for Overlapping Address Pool, page 2-30

The information provided here applies only to dial-in access to MPLS VPN integration. For more information about the configuration and troubleshooting tasks associated with direct dial-in, please refer to *Configuring Virtual Private Networks* at the following URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/dialns_c/dnsprt3/dcdvpn.htm

show Commands for NAS/PE

- **show user—**Displays information about the active lines on the router per a specific user.
- **show caller—**Displays individual users and consumed resources on the NAS, and active call statistics for large pools of connections, and the absolute and idle times for each user.
- **show caller user** *caller name***—**Displays the show caller command information for a particular user.
- **show resource-pool call—**(RPMS-specific) Displays all active call information for all customer profiles and resource groups.
- **show resource-pool resource—**(RPMS-specific) Displays resource groups configured in the network access server.

- **show interface virtual-access**—Displays detailed information about the virtual access interface.
- **show virtual access configuration**—Displays detailed information about the virtual access interface configuration.
- **show ip route**—Displays the current state of the routing table, including the IP address, the network mask, protocol, and static routes.
- **show ip local pool**—Displays the address pools that have been downloaded to a Cisco network access server.

Example 2-26 shows the detailed output that results when you implement these show commands.

Example 2-26 Sample show Command Output for NAS/PE

show Commands for Overlapping Address Pool

The overlapping address pool feature allows the administrator to configure overlapping IP address pool groups in order to concurrently use the IP addresses in multiple address spaces.

The design of this feature ensures that a pool name is an explicit selection of a group, or address space. There are, however, a few characteristics about this feature that the administrator must know.

With Overlapping Address Pool, you *can*:

- Assign pools in different groups to a single address space.
- Use with applications that cannot handle duplicate address assignments in their routing table.
- Use on all DSL and Dial platforms that support the current IP local address pools.
- On the Dial platforms, PPP over ISDN sessions will try to use overlapping IP address pools with MPLS VPN configuration.

With Overlapping Address Pool, you *cannot*:

- Associate a pool name with more than one group.
- Associate the pool name default with a named group. It can be used only in the BASE group.

Note The overlapping address feature is independent of the type of configuration used on the dialer.

To verify that an IP address is active, use the **show local pool** command in user EXEC mode. The output shown in Example 2-27 tells you which IP address has been received, and that the call is active and in use.

Example 2-27 Sample show Command Output for Overlapping Address Pool

```
c72c3-1# sh ip local pool V1.28-pool

 Pool                     Begin           End             Free  In use
 ** pool <V1.28-pool> is in group <V1.28-group>
 V1.28-pool               10.1.28.1       10.1.28.10         9       1
Available addresses:
   10.1.28.5
   10.1.28.6
   10.1.28.7
   10.1.28.8
   10.1.28.9
   10.1.28.10
   10.1.28.1
   10.1.28.2
   10.1.28.3
Inuse addresses:
   10.1.28.4          Vi9                         U0001N3P1@V1.28.com
```

Troubleshooting Dial Backup

Dial backup is used as a backup for direct remote connections such as cable or DSL. Using L2TP dial-in, it provides backup connectivity from the customer's remote office to the customer's VPN when their primary link becomes unavailable.

The primary link and the backup link are typically configured on the same CE router at the remote site.

Understanding Dial Backup Call Flow

Call flow in dial backup is identical to that in L2TP dial-in access, except that the call is initiated by a backup interface, when connectivity to the primary interface is lost, instead of by a remote user. The backup interface is a dialer interface configured to dial into the service provider's NAS on a dial backup phone number. (The phone number indicates that dial backup is being initiated instead of a typical L2TP dial-in.)

Using L2TP, the NAS tunnels the PPP session to the VHG/PE, which then maps the incoming session into the appropriate VRF.

When the primary link is restored, the primary route will also be restored, the backup connection terminated by the remote user, and the backup route deleted by the VHG/PE.

This is depicted in Figure 2-8.

Figure 2-8 Topology of Dial Backup

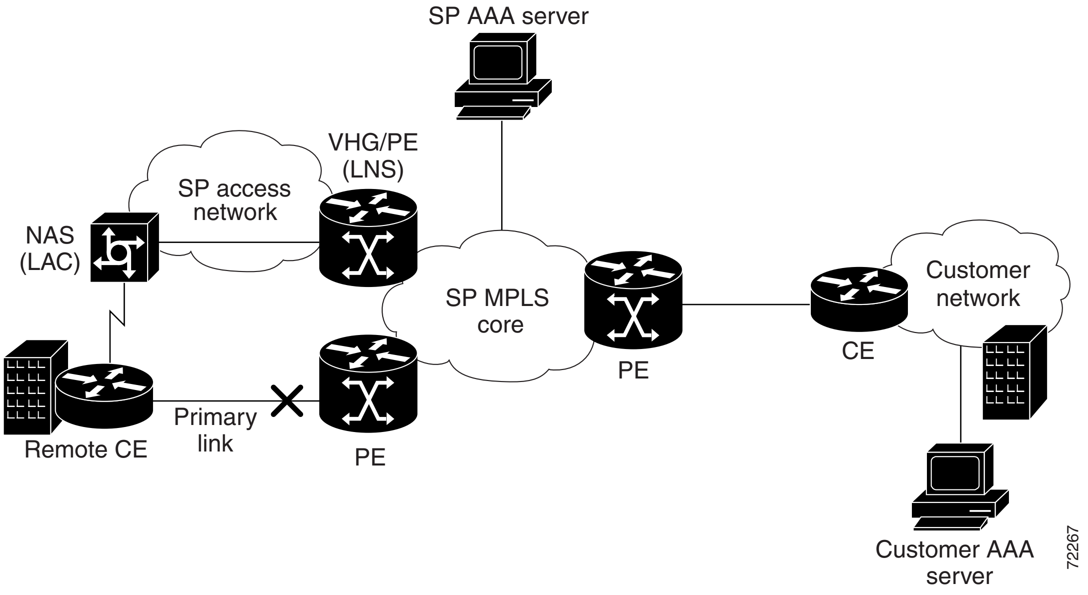


Table 2-3 describes the major dial backup events.

Table 2-3 Troubleshooting Dial Backup Call Flow Events

Step in Call Flow	Troubleshooting Topic
1. Connectivity is lost (for a specified time) on the primary connection. On the remote CE, the backup interface is activated and dials a backup phone number on the NAS. The customer's subnets are moved from the primary link to the backup link.	Initiating Backup Connection, Switching Call to Backup Link, page 2-32
2. The session proceeds as in L2TP dial-in access to MPLS VPN. The NAS tunnels the PPP (or MLP) session to the VHG/PE, which then maps the incoming session into the appropriate VRF. Connectivity is maintained and the remote user is again part of the customer VPN. Packets can flow from/to the remote user.	Once you verify that the backup link is dialed, troubleshoot as in troubleshooting normal L2TP dial-in access.
3. When the primary link is restored, the backup connection is terminated by the remote user, the VHG/PE deletes the backup route, and the CE switches back to the primary link.	Terminating Backup Connection, Switching Call to Primary Link, page 2-33

Procedure for Troubleshooting Call Flow in Dial Backup

Dial backup problems can occur either in the first step in the call flow, when the primary link goes down and the dial backup link is to be dialed, or in the last step, when the primary link comes back up and the backup link is to be deactivated. This section describes procedures for troubleshooting these steps, followed by an example of the output of the commands used.

Initiating Backup Connection, Switching Call to Backup Link

Is the backup link dialed when the primary link goes down?

- Check that when the primary link goes down, the interface on which the **backup interface** command is configured goes down as well. For example, if the primary interface is interface Serial 0, then the line protocol for that interface must go down for the backup interface to be brought out of standby. Since the backup interface method relies on the interface it is configured on to be in a down state before the backup interface actually comes up, verify that a primary link failure is actually reflected in the state of the interface. You can determine the state of the interface using the command **show interface** *interface slot/port*. To determine the dial backup interface, use the command **show backup**.
- Check to see if the router generated a console message indicating that the backup interface changed out of standby mode. This message will only appear after the enable-timer, specified by the backup delay enable-timer disable-timer command, has expired. If you do not see this console message, adjust the backup delay enable timer to a lower value. An example of a 10 second delay timer is shown here:

```
*Mar  1 03:37:31.788: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0,
      changed state to down
!-- The primary interface goes down
*Mar  1 03:37:42.719: %LINK-3-UPDOWN: Interface Dialer1, changed state to up
```

```
!-- The backup interface is brought out of standby mode approximately
!-- ten seconds later
```

- Verify the routing table contains a valid route to the backup interface to be dialed. Use the **show ip route** command to verify that the route exists in the routing table after the primary link goes down. The floating static route will only be installed in the routing table after all other identical routes, with lower administrative distance are removed. Check to make sure that there are no other sources for the primary route (possibly due to a routing loop).
- Check that the interesting traffic definition is correctly defined (using the dialer-list command) and is applied to the interface (using the dialer-group command) providing the backup. Generate interesting traffic, then use the command **debug dialer packet** to verify the traffic is designated interesting and can bring up the link.

Note The routing protocol should not be defined as interesting. This prevents the periodic updates or hellos from keeping the backup link up indefinitely. The following is an example of a good interesting traffic definition for this backup method:

Terminating Backup Connection, Switching Call to Primary Link

Is the backup link deactivated when the primary link recovers?

- Check that when the primary link recovers, the interface (on which the **backup interface** command is configured) comes up as well. This is necessary since the router will not recognize that the primary link is up until the line protocol of that interface is up. For example, if the primary interface is interface Serial 0, then the line protocol for that interface must come up for the backup interface to change into standby. You can determine the state of the interface using the command **show interface** *interface slot/port.* To determine the dial backup interface, use the command **show backup**.
- Verify that the disable timer is set appropriately. The disable timer is specified with the command **backup delay** *enable-timer disable-timer.* For example, the command **backup delay 10 60** indicates that the backup link will be enabled 10 seconds after the primary link goes down, and that the backup link will be brought down 60 seconds after the primary link recovers. If your backup link stays up longer than desired, adjust the disable time downwards.
- Use **show ip route** to verify that the routing protocol reinstalls the primary route. This should cause the floating static route to be removed from the routing table. All traffic should now use the primary link. If the primary route is not reinstalled, troubleshoot the routing protocol.
- Use **debug dialer** to verify that there is no interesting traffic that passes on the backup link. Since interesting traffic resets the idle timeout, the link will not be brought down if there is unwanted interesting traffic. Watch for certain broadcast and multicast packets that can reset the idle time-out. If necessary, modify the interesting traffic definition to be more restrictive and designate such rogue packets as not interesting.
- Lower the dialer idle-timeout (default is 120 seconds). Because the backup link is only brought down when the idle time-out expires, a lower idle timeout can bring down the backup link.

Example 2-28 Sample Results of Commands on Remote CE

```
CE router# show debug

Backup:
  Backup events debugging is on
```

```
PPP:
  PPP authentication debugging is on
  PPP protocol errors debugging is on
  PPP protocol negotiation debugging is on
IP routing:
  IP routing debugging is on

CE router# show backup

Primary Interface   Secondary Interface   Status
-----------------   -------------------   ------
Serial7:0           Dialer81              normal operation

CE router# show interface se7:0
Serial7:0 is up, line protocol is up
  Hardware is DSX1
  Internet address is 10.1.8.5/30
  Backup interface Dialer81, failure delay 0 sec, secondary disable delay 0 sec,

/*'backup delay {enable-delay|never} {disable-delay|never}'
  enable-delay   Number of seconds that elapse after the primary line goes down
                 but before IOS activates the secondary line.
  disable-delay  Number of seconds that elapse after the primary line comes up
                 but before IOS deactivates the secondary line.
  never          Prevents secondary line from being activated or deactivated.*/

  kickin load not set, kickout load not set

/*'backup load {enable-threshold|never} {disable-threshold|never}'
  enable-threshold  Percentage of the primary line's available bandwidth
                    that the traffic load must exceed before to enable
                    dial backup.
  disable-threshold Percentage of the primary line's available bandwidth
                    that the load must be less than to disable dial backup.
  never             Set the secondary line never to be activated due to
                    traffic load.*/


  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:01, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     75 packets input, 10526 bytes, 0 no buffer
     Received 52 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5500 packets output, 137397 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
  Timeslot(s) Used:1, Transmitter delay is 0 flags

CE router# show dialer interface dialer 81

Di81 - Dialing not enabled on this interface.

CE router# show ip route 10.1.8.241

Routing entry for 10.1.8.241/32
```

```
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 10.1.8.6 on Serial7:0, 00:00:26 ago
  Routing Descriptor Blocks:
  * 10.1.8.6, from 10.1.8.6, 00:00:26 ago, via Serial7:0
      Route metric is 1, traffic share count is 1
```

...The primary link goes down...

```
Jul  4 08:20:00.042: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial7:0, changed 
state to down
Jul  4 08:20:00.042: BACKUP(Serial7:0): event = primary went down
Jul  4 08:20:00.042: BACKUP(Serial7:0): changed state to "waiting to backup"
Jul  4 08:20:00.042: RT: del 10.2.8.241/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.042: RT: delete subnet route to 10.2.8.241/32
Jul  4 08:20:00.042: RT: del 10.2.8.242/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.042: RT: delete subnet route to 10.2.8.242/32
Jul  4 08:20:00.042: RT: del 10.2.8.243/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.042: RT: delete subnet route to 10.2.8.243/32
Jul  4 08:20:00.042: RT: del 10.2.8.244/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.042: RT: delete subnet route to 10.2.8.244/32
Jul  4 08:20:00.042: RT: del 10.2.8.245/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.042: RT: delete subnet route to 10.2.8.245/32
Jul  4 08:20:00.046: RT: del 10.2.8.246/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.046: RT: delete subnet route to 10.2.8.246/32
Jul  4 08:20:00.046: RT: del 10.2.8.247/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.046: RT: delete subnet route to 10.2.8.247/32
Jul  4 08:20:00.046: RT: del 10.2.8.248/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.046: RT: delete subnet route to 10.2.8.248/32
Jul  4 08:20:00.046: RT: del 10.2.8.249/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.046: RT: delete subnet route to 10.2.8.249/32
Jul  4 08:20:00.046: RT: del 10.1.8.241/32 via 10.1.8.6, rip metric [120/1]
Jul  4 08:20:00.046: RT: delete subnet route to 10.1.8.241/32
Jul  4 08:20:00.046: RT: del 10.1.8.242/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.046: RT: delete subnet route to 10.1.8.242/32
Jul  4 08:20:00.046: RT: del 10.1.8.245/32 via 10.1.8.6, rip metric [120/1]
Jul  4 08:20:00.046: RT: delete subnet route to 10.1.8.245/32
Jul  4 08:20:00.050: RT: del 10.1.8.246/32 via 10.1.8.6, rip metric [120/9]
Jul  4 08:20:00.050: RT: delete subnet route to 10.1.8.246/32

Jul  4 08:20:00.050: BACKUP(Serial7:0): event = timer expired
Jul  4 08:20:00.050: BACKUP(Serial7:0): secondary interface (Dialer81) made active
Jul  4 08:20:00.050: BACKUP(Serial7:0): changed state to "backup mode"

Jul  4 08:20:00.150: RT: interface Serial7:0 removed from routing table
Jul  4 08:20:00.150: RT: del 10.1.8.4/30 via 0.0.0.0, connected metric [0/0]
Jul  4 08:20:00.150: RT: delete subnet route to 10.1.8.4/30
Jul  4 08:20:00.154: RT: add 10.1.8.0/30 via 0.0.0.0, connected metric [0/0]
Jul  4 08:20:00.154: RT: interface Dialer81 added to routing table
Jul  4 08:20:01.154: RT: add 10.2.8.250/32 via 0.0.0.0, static metric [220/0]
Jul  4 08:20:01.154: RT: add 10.1.8.0/24 via 0.0.0.0, static metric [230/0]

Jul  4 08:20:02.050: %LINK-3-UPDOWN: Interface Dialer81, changed state to up

Jul  4 08:20:02.050: Di81 LCP: Not allowed on a Dialer Profile
Jul  4 08:20:02.050: BACKUP(Dialer81): event = primary came up

CE router# show ip route 10.1.8.241

Routing entry for 10.1.8.0/24
```

```
  Known via "static", distance 230, metric 0 (connected)
  Redistributing via rip
  Advertised by rip
  Routing Descriptor Blocks:
  * directly connected, via Dialer81
      Route metric is 0, traffic share count is 1

CE router# show inter se7:0

Serial7:0 is up, line protocol is down
  Hardware is DSX1
  Internet address is 10.1.8.5/30
  Backup interface Dialer81, failure delay 0 sec, secondary disable delay 0 sec,
  kickin load not set, kickout load not set
  MTU 1500 bytes, BW 64 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:56, output never, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  30 second input rate 0 bits/sec, 0 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
     86 packets input, 11737 bytes, 0 no buffer
     Received 57 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     5519 packets output, 138641 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions
  Timeslot(s) Used:1, Transmitter delay is 0 flags

CE router# show dialer interface dialer 81

Di81 - dialer type = DIALER PROFILE
Idle timer (120 secs), Fast idle timer (300 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Number of active calls = 1
Dial String      Successes   Failures    Last DNIS   Last status
8110                     3          0    00:01:26       successful   Default
Serial0:15 - dialer type = ISDN
Dial String      Successes   Failures    Last DNIS   Last status
0 incoming call(s) have been screened.
0 incoming call(s) rejected for callback.
Serial0:0 - dialer type = ISDN
Idle timer (120 secs), Fast idle timer (300 secs)
Wait for carrier (30 secs), Re-enable (15 secs)
Dialer state is data link layer up
Dial reason: ip (s=10.1.8.1, d=10.1.8.241)
Interface bound to profile Di81
Time until disconnect 35 secs
Current call connected 00:01:28
Connected to 8110 (8110)

CE router# show backup

Primary Interface   Secondary Interface   Status
-----------------   -------------------   ------
Serial7:0           Dialer81              backup mode

CE router# show ip route 10.1.8.241
```

```
Routing entry for 10.1.8.241/32
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 10.1.8.2 on Dialer81, 00:00:20 ago
  Routing Descriptor Blocks:
  * 10.1.8.2, from 10.1.8.2, 00:00:20 ago, via Dialer81
      Route metric is 1, traffic share count is 1
```

To summarize, ping through the MPLS network, from end to end.

```
CE router# ping 10.1.8.241

Jul  4 08:20:15.422: %LINK-3-UPDOWN: Interface Serial0:0, changed state to up
Jul  4 08:20:15.422: Se0:0 PPP: Phase is DOWN, Setup [0 sess, 0 load]
Jul  4 08:20:15.422: Se0:0 PPP: Treating connection as a callout
Jul  4 08:20:15.422: Se0:0 PPP: Phase is ESTABLISHING, Active Open [0 sess, 0 load]
Jul  4 08:20:15.426: Se0:0 PPP: No remote authentication for call-out
Jul  4 08:20:15.426: Se0:0 LCP: O CONFREQ [Closed] id 1 len 10
Jul  4 08:20:15.426: Se0:0 LCP:    MagicNumber 0x53B5CC6E (0x050653B5CC6E)
Jul  4 08:20:15.426: %DIALER-6-BIND: Interface Se0:0 bound to profile Di81
Jul  4 08:20:15.426: Se0:0 PPP: Treating connection as a callout
Jul  4 08:20:15.434: Se0:0 LCP: I CONFREQ [REQsent] id 30 len 15
Jul  4 08:20:15.434: Se0:0 LCP:    AuthProto CHAP (0x0305C22305)
Jul  4 08:20:15.434: Se0:0 LCP:    MagicNumber 0xEF83FEAF (0x0506EF83FEAF)
Jul  4 08:20:15.434: Se0:0 LCP: O CONFACK [REQsent] id 30 len 15
Jul  4 08:20:15.434: Se0:0 LCP:    AuthProto CHAP (0x0305C22305)
Jul  4 08:20:15.434: Se0:0 LCP:    MagicNumber 0xEF83FEAF (0x0506EF83FEAF)
Jul  4 08:20:15.438: Se0:0 LCP: I CONFACK [ACKsent] id 1 len 10
Jul  4 08:20:15.438: Se0:0 LCP:    MagicNumber 0x53B5CC6E (0x050653B5CC6E)
Jul  4 08:20:15.438: Se0:0 LCP: State is Open
Jul  4 08:20:15.438: Se0:0 PPP: Phase is AUTHENTICATING, by the peer [0 sess, 0 load]
Jul  4 08:20:15.446: Se0:0 CHAP: I CHALLENGE id 30 len 28 from "c53d9-1"
Jul  4 08:20:15.446: Se0:0 CHAP: Using alternate hostname U0001N1P3V1.8@V1.8.com
Jul  4 08:20:15.446: Se0:0 CHAP: Username c53d9-1 not found
Jul  4 08:20:15.446: Se0:0 CHAP: Using default password
Jul  4 08:20:15.446: Se0:0 CHAP: O RESPONSE id 30 len 43 from "U0001N1P3V1.8@V1.8.com"
Jul  4 08:20:15.490: Se0:0 CHAP: I SUCCESS id 30 len 4
Jul  4 08:20:15.490: Se0:0 PPP: Phase is UP [0 sess, 0 load]
Jul  4 08:20:15.490: Se0:0 IPCP: O CONFREQ [Closed] id 1 len 10
Jul  4 08:20:15.490: Se0:0 IPCP:    Address 10.1.8.1 (0x030620010801)
Jul  4 08:20:15.522: Se0:0 IPCP: I CONFREQ [REQsent] id 1 len 10
Jul  4 08:20:15.522: Se0:0 IPCP:    Address 10.1.8.2 (0x030620010802)
Jul  4 08:20:15.522: Se0:0 IPCP: O CONFACK [REQsent] id 1 len 10
Jul  4 08:20:15.522: Se0:0 IPCP:    Address 10.1.8.2 (0x030620010802)
Jul  4 08:20:15.526: Se0:0 IPCP: I CONFACK [ACKsent] id 1 len 10
Jul  4 08:20:15.526: Se0:0 IPCP:    Address 10.1.8.1 (0x030620010801)
Jul  4 08:20:15.526: Se0:0 IPCP: State is Open
Jul  4 08:20:15.526: RT: add 10.1.8.2/32 via 0.0.0.0, connected metric [0/0]
Jul  4 08:20:15.526: Di81 IPCP: Install route to 10.1.8.2
Jul  4 08:20:16.490: %LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0:0, changed
state to up
Jul  4 08:20:17.866: RT: add 10.2.8.241/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.866: RT: add 10.2.8.242/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.866: RT: add 10.2.8.243/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.866: RT: add 10.2.8.244/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.866: RT: add 10.2.8.245/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.870: RT: add 10.2.8.246/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.870: RT: add 10.2.8.247/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.874: RT: add 10.2.8.248/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.874: RT: add 10.2.8.249/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.874: RT: add 10.1.8.241/32 via 10.1.8.2, rip metric [120/1]
Jul  4 08:20:17.878: RT: add 10.1.8.242/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:17.878: RT: add 10.1.8.245/32 via 10.1.8.2, rip metric [120/1]
Jul  4 08:20:17.878: RT: add 10.1.8.246/32 via 10.1.8.2, rip metric [120/9]
Jul  4 08:20:21.426: %ISDN-6-CONNECT: Interface Serial0:0 is now connected to 8110 unknown
```

Troubleshooting Dial-out Access

In dial-out remote access, instead of a remote user or CE initiating a call into the MPLS VPN, the connection is established by traffic coming *from* the MPLS VPN and triggering a call from the dial-out router to the remote CE. Dial-out access can use either L2TP or direct ISDN architecture.

Dial-out is often used for automated functions. For example, a central database system might dial out nightly to remote vending machines to collect daily sales data and check inventories.

In this release of Cisco Remote Access to MPLS VPN integration, the dialer interface used is a *dialer profile*. With a dialer profile, each physical interface becomes a member of a dialer pool. The VHG/PE (in L2TP dial-out) or the NAS/PE (in direct dial-out) triggers a call when it receives interesting traffic from a remote peer in the customer VPN. ("Interesting traffic" is traffic identified as destined for this particular dial-out network.)

Based on the dialer interface configuration, the VHG/PE or NAS/PE borrows a physical interface from the dialer pool for the duration of the call. Once the call is complete, the router returns the physical interface to the dialer pool. Because of this dynamic binding, different dialer interfaces can be configured for different customer VPNs, each with its own VRF, IP address, and dialer string.

Unlike dial-in remote access, dial-out access does not require the querying of an AAA server or the use of two-way authentication, because user information is directly implemented on the dialer profile interface configured on the dial-out router.

Dial-out access may include these features:

- Multilink PPP (MLP)—PPP that is split across multiple data links.
- Multichassis MLP (MMP)—MLP with redundant "stacked" NAS/PEs, using a stack group bidding process (SGBP) to manage the allocation of PPP sessions among the members of the stack.
- Address management through overlapping local pools configured on the NAS/PE or overlapping address pools on the SP AAA server, through a DHCP (dynamic host configuration protocol) server, or through on-demand address pools (ODAP).

Figure 2-9 shows an example of the topology for L2TP dial-out access, and Figure 2-10 shows an example of the topology for direct ISDN PE dial-out access.

Figure 2-9 Topology of L2TP Dial-out Remote Access

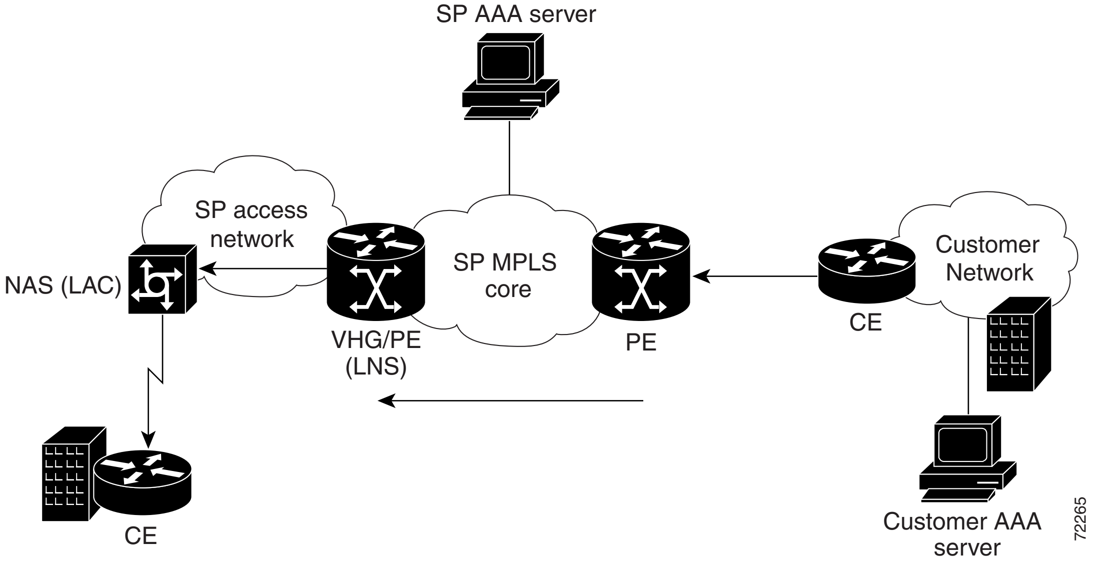
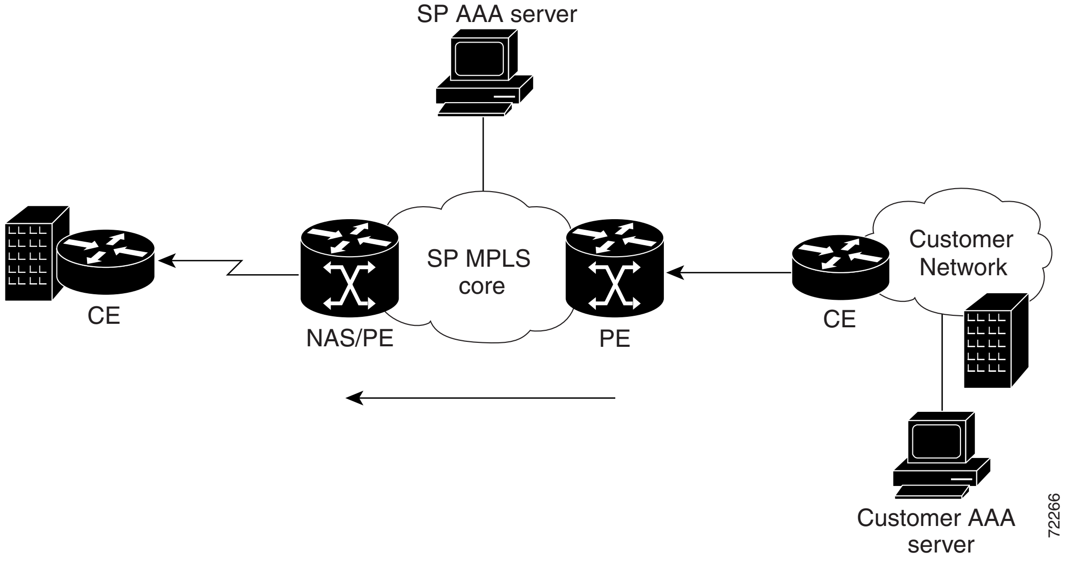


Figure 2-10 Topology of Direct ISDN PE Dial-out Remote Access



These are the main events in the dial-out call flow:

1. Traffic from a specific customer VPN, destined for a specific dial-out network (identified through static routes in the customer VRF) is directed to the appropriate VHG/PE or NAS/PE.
2. Upon receiving the traffic, either the VHG/PE or the NAS/PE responds:
   - In L2TP dial-out, the VHG/PE brings up an L2TP tunnel and negotiates an outgoing PPP session with the NAS. The dial-out PPP session is triggered using dialer profiles. The NAS then dials out to the CE using dial-out information received in the L2TP session negotiation.

- In direct dial-out, the NAS/PE dials out directly to the CE. The dial-out PPP session is triggered using dialer profiles.

Troubleshooting L2TP Dial-out Access

The following troubleshooting process can be used to resolve common dial-out scenarios in this solution. Exhaustive troubleshooting of all possible combinations of features is beyond the scope of this document.

Part 1: Before the Call Has Been Brought Up

Follow these steps to troubleshoot call flow before the call has been brought up.

Step 1. Is the VHG/PE Receiving Traffic?

Is the VHG/PE receiving traffic for the remote CE? Check to see if a static route is configured on the VHG/PE. In the following example, the static route has been created for Dialer50.

Example 2-29 Checking for a Static Route

```
router# sh ip route vrf V1.17.com static
     10.0.0.0/32 is subnetted, 6 subnets
S       10.1.17.40 is directly connected, Dialer53
S       10.1.17.30 is directly connected, Dialer52
S       10.1.17.20 is directly connected, Dialer51
S       10.1.17.10 is directly connected, Dialer50
```

Is the route being forwarded to the remote PEs?

If the static route has been created, you can check for more route-specific detail, including whether the route is being forwarded via MP-BGP to other remote PEs:

Example 2-30 Checking Route-Specific Detail

```
router# sh ip route vrf V1.17.com 10.1.17.10
Routing entry for 10.1.17.10/32
  Known via "static", distance 1, metric 0 (connected)
  Redistributing via bgp 100  <<<
  Advertised by bgp 100       <<<
  Routing Descriptor Blocks:
  * directly connected, via Dialer50, permanent
      Route metric is 0, traffic share count is 1
```

On the remote PE, is the route reflected in the routing table?

If you determine that the route was forwarded, check the remote PE to make sure the static route is reflected in the routing table for this VRF. In the example below, the remote PE is aware that traffic for 10.1.17.10 (the remote CE IP address) should be forwarded to the VHG/PE 10.10.104.12:

Example 2-31 Checking the Remote PE

On the remote PE:

```
router# sh ip route vrf V1.17.com 10.1.17.10
Routing entry for 10.1.17.10/32
  Known via "bgp 100", distance 200, metric 1, type internal
  Last update from 10.10.104.12 00:23:07 ago
  Routing Descriptor Blocks:
  * 10.10.104.12 (Default-IP-Routing-Table), from 10.10.104.12, 00:23:07 ago
      Route metric is 1, traffic share count is 1
      AS Hops 0
```

Step 2. Is VRF FIB Information Forwarded in the MPLS Core?

To properly forward traffic to the VHG/PE, the remote PE must know not only the IP address of the remote CE but also which outgoing interface the VHG/PE should use for this CE.

If the route is properly reflected in the remote PE's routing table, check the VHG/PE configuration for the correct MPLS path. Specifically, make sure that the MPLS forwarding table includes the tag that the remote CE must use when forwarding traffic to the remote CE via this VHG/PE:

On the VHG/PE, is the MPLS path configured correctly?

Example 2-32 Checking MPLS Path

```
router# show mpls forwarding-table vrf V1.17.com 10.1.17.10
Local  Outgoing    Prefix       Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id  switched   interface
82     Aggregate   10.1.17.10/32[V]  0
```

Note that at this point, the outgoing interface information is missing. The CEF switching path is not fully complete until the call is brought up.

On the remote PE, are the necessary tags appearing?

Check that the CEF entry for the remote CE (10.1.17.10) includes the two required tags, one to reach the VHG/PE (its IP address, 10.10.104.12) and another for the remote CE IP address (10.1.17.10) entry in the V1.17.com VRF table:

Example 2-33 Checking CEF

```
router# sh ip cef vrf V1.17.com 10.1.17.10
10.1.17.10/32, version 163, cached adjacency 10.10.103.113
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et0/1, 10.10.103.113, tags imposed: {78 82}
via 10.10.104.12, 0 dependencies, recursive
  next hop 10.10.103.113, Ethernet0/1 via 10.10.104.12/32
valid cached adjacency
  tag rewrite with Et0/1, 10.10.103.113, tags imposed: {78 82}
```

In this example, *78* is the tag to reach 10.10.104.12, the VHG/PE. *82* is the tag for selecting the correct outgoing interface on the VHG/PE.

If the tags do not appear on the remote PE, there is a problem in the MPLS VPN core. Troubleshooting MPLS VPN is beyond the scope of this document. For more information, see the MPLS Troubleshooting Guide.

Step 3. Is CEF Switching Configured to Trigger the Call?

In Step 2 above, the outgoing interface in the show mpls command on the VHG/PE was not yet filled in. When a packet comes in from the remote PE with a tag header 82 – the tag for selecting the outgoing interface – the VHG/PE does not yet know what interface to use.

On the VHG/PE, the incoming packet will be processed by the CEF switching path. In this step, check that CEF switching entries have been defined to trigger the call. In the example, the CEF switching path for 10.1.17.10 (the CE) shows that the IP CEF switching path has a valid CEF entry to reach that address and Dialer50.

This entry makes it possible to trigger the call because the tag 82 in the VRF FIB table shows it is linked to a route for 10.1.17.10, which is in the VRF FIB as an *aggregate*.

With an aggregate route, CEF lookup is not done via the CEF fast switching path and information in the adjacency table, but is handled by the IP CEF process switching routine. Packets are handed off to the dialer routing code.

Note Because the call is not yet established, the CEF entry is still incomplete. The tag rewrite is empty and there is no reference to the adjacency for the outgoing interface. When the call is established, this information will be completed.

Example 2-34 Checking CEF Switching

```
router# sh ip cef vrf V1.17.com 10.1.17.10
10.1.17.10/32, version 9, epoch 0, attached
0 packets, 0 bytes
  tag information set
    local tag: 82
  via Dialer50, 0 dependencies
    valid punt adjacency
    tag rewrite with , , tags imposed: {} (***see note below)
c72d2-2# sh ip cef vrf V1.17.com dialer 50
Prefix              Next Hop             Interface
10.1.17.10/32       attached             Dialer50
```

Part 2. After the Call Is Brought Up

Step 1. Is the Call Triggered and a PPP Connection Established?

With CEF switching properly configured on the VHG/PE, if a packet comes in on the ingress MPLS interface with a tag header containing, in this example, tag value 82, the VHG/PE looks at the IP CEF switching path in the VRF and determined that it must go on dialer 50.

At this point the IP packet is punted [is there a better word?] to the dialer code. The dialer routine triggers the call and brings up the PPP connection to the remote CE. As soon the connection is up, the MPLS and CEF switching procedures will adjust to the new changes.

On the VHG/PE, is the route connected?

In this example, the route is connected (the route metric is changed from static to connected):

Example 2-35 Checking Route Connection

```
router# sh ip route vrf V1.17.com connected
    10.0.0.0/32 is subnetted, 6 subnets
    10.1.17.10 is directly connected, Dialer50
```

On the VHG/PE, is the show mpls information updated?

In this example, the show mpls information is updated for tag 82. Note that the tag relationship has changed from *aggregate* to *untagged*, and the outgoing interface uses vi2:

Example 2-36 Checking MPLS Update

```
router# sh mpls forwarding-table vrf V1.17.com 10.1.17.10
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
82     Untagged    10.1.17.10/32[V]  722072     Vi2        point2point
```

Is the CEF switching path updated?

With the change from aggregate to untagged in the show mpls information above, CEF lookup is no longer done via the IP CEF process switching path.

With an outgoing interface (vi2) stipulated, the CEF lookup process will now look for adjacencies for this outgoing interface. CEF fast switching is in place, as shown in the following example:

Example 2-37 Checking CEF Switching Path Update

```
router# sh adj virtual-acc 2
Protocol Interface                 Address
TAG      Virtual-Access2           point2point(4) (incomplete)
IP       Virtual-Access2           point2point(33)
router# sh adj virtual-acc 2 detail
Protocol Interface                 Address
TAG      Virtual-Access2           point2point(4) (incomplete)
                                   0 packets, 0 bytes
                                   mpls adj   never
                                   Epoch: 0
IP       Virtual-Access2           point2point(33)
                                   8098 packets, 842192 bytes
                                   FF030021
                                   Epoch: 0
```

The CEF lookup path is changed in order to process the packets faster. Packets are now processed by checking the adjacency entries.

Note Although the TAG path shows incomplete, this is normal since the tag header of the incoming MPLS packet should not be forwarded over the PPP link. Since the TAG path is incomplete, it will now look at the IP adjacency when forwarding the IP header and data received from the MPLS cloud and append the header "FF030021" in front of it.

Debugging Commands for L2TP Dial-out

For general dial-out troubleshooting, the following debug commands may be used. Debug commands are issued in enable mode. Debug output examples are shown in Table 2-4.

Table 2-4 debugging Commands for Dial-out

Command	Use To
debug dialer	Show information on the dialer profile.

Table 2-4 debugging Commands for Dial-out (continued)

Command	Use To
deb ip cef dialer	Show the change in the CEF switching path, as discussed in Part 2 above.
deb mpls pac	Show information on the mpls packet. **Caution** In a production environment, this command can produce much more output than the packet you are interested in.

Debugging Examples

This is the incoming MPLS packet with a tag header containing tag value 82, destined for remote CE and triggering the call. The following debugging shows this.

```
router#
Mar  7 09:58:37.878: TAG: PO5/0: recvd: CoS=0, TTL=252, Tag(s)=82
Mar  7 09:58:37.882: Vi31 DDR: Dialing cause ip (s=10.2.17.241, d=10.1.17.10)
Mar  7 09:58:37.882: Vi31 DDR: Attempting to dial 11710
Mar  7 09:58:37.914: %LINK-3-UPDOWN: Interface Virtual-Access31, changed state to up
Mar  7 09:58:37.914: Vi31 DDR: Dialer statechange to up
Mar  7 09:58:37.914: %DIALER-6-BIND: Interface Vi31 bound to profile Di50
Mar  7 09:58:37.914: Vi31 DDR: Dialer call has been placed
Mar  7 09:58:37.966: %DIALER-6-BIND: Interface Vi30 bound to profile Di50
Mar  7 09:58:37.970: %LINK-3-UPDOWN: Interface Virtual-Access30, changed state to up
Mar  7 09:58:37.970: Vi30 DDR: Dialer statechange to up
Mar  7 09:58:37.970: Vi30 DDR: Dialer call has been placed
```

Debugging after changes in the CEF switching path: The following output is seen only if the call comes up. If the call does not come up, troubleshooting is the same as for a non-MPLS call.

Example 2-38 debug Output When the Call Is Up

```
Mar  7 09:58:37.978: CEF-Dialer (legacy): add link to 10.1.17.10 via Dialer50 through
Virtual-Access30
Mar  7 09:58:37.978: CEF-Dialer: adjacency added: 0x62F77AC0
Mar  7 09:58:37.978: CEF-Dialer: adjacency found: 0x62F77AC0; fib->count: 1
Mar  7 09:58:37.978: CEF-Dialer: setup loadinfo with 1 paths
Mar  7 09:58:37.978: Vi30 DDR: dialer protocol up
Mar  7 09:58:38.958: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access31,
changed state to up
Mar  7 09:58:38.970: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access30,
changed state to up
```

Example 2-39 debug Output When the Call Goes Down

```
deb ip cef dialer
Mar  7 12:41:47.853: CEF-Dialer (legacy): remove link to 10.1.17.10 via Dialer50 through
Virtual-Access30
```

Troubleshooting Direct ISDN PE Dial-out

The following troubleshooting process can be used to troubleshoot common dial-out scenarios in this solution, which uses direct ISDN PE dial-out with dialer profiles. Exhaustive troubleshooting of all possible combinations of features is beyond the scope of this document.

Part 1: Before the Call Has Been Brought Up

Step 1. Is the NAS/PE Receiving Traffic?

Is the NAS/PE receiving traffic for the remote CE?

Check to see if a static route is configured on the NAS/PE.

In the following example, the static route has been created for Dialer50:

Example 2-40 show ip route

```
router# sh ip route vrf V1.17.com static
     10.0.0.0/32 is subnetted, 6 subnets
S       10.1.17.40 is directly connected, Dialer53
S       10.1.17.30 is directly connected, Dialer52
S       10.1.17.20 is directly connected, Dialer51
S       10.1.17.10 is directly connected, Dialer50
```

Is the route being forwarded to remote PEs?

If the static route has been created, you can check for more route-specific detail, including whether the route is being forwarded via MP-BGP to other remote PEs:

Example 2-41 show ip route detail

```
router# sh ip route vrf V1.17.com 10.1.17.10
Routing entry for 10.1.17.10/32
  Known via "static", distance 1, metric 0 (connected)
  Redistributing via bgp 100  <<<
  Advertised by bgp 100       <<<
  Routing Descriptor Blocks:
  * directly connected, via Dialer50, permanent
      Route metric is 0, traffic share count is 1
```

On the remote PE, is the route reflected in the routing table?

If you determine that the route was forwarded, check the remote PE to make sure the static route is reflected in the routing table for this VRF. In the example below, the remote PE is aware that traffic for 10.1.17.10 (the remote CE IP address) should be forwarded to the VHG/PE 10.10.104.12:

On the remote PE, enter the **show ip route vrf *vpn*** command, as in this example:

Example 2-42 show ip route vrf

```
router# sh ip rout vrf V1.17.com 10.1.17.10
Routing entry for 10.1.17.10/32
  Known via "bgp 100", distance 200, metric 1, type internal
  Last update from 10.10.104.12 00:23:07 ago
  Routing Descriptor Blocks:
  * 10.10.104.12 (Default-IP-Routing-Table), from 10.10.104.12, 00:23:07 ago
      Route metric is 1, traffic share count is 1
      AS Hops 0
```

Step 2. Is VRF FIB Information Forwarded in the MPLS Core?

To properly forward traffic to the NAS/PE, the remote PE must know not only the remote CE's IP address but also which outgoing interface the NAS/PE should use for this CE.

If the route is properly reflected in the remote PE's routing table, check the NAS/PE configuration for the correct MPLS path. Specifically, make sure that the MPLS forwarding table includes the tag that the remote CE must use when forwarding traffic to the remote CE via this NAS/PE:

On the NAS/PE, is the MPLS path configured correctly?

Example 2-43 show mpls forwarding-table

```
router# show mpls forwarding-table vrf V1.17.com 10.1.17.10
Local  Outgoing    Prefix       Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id  switched   interface
82     Aggregate   10.1.17.10/32[V]  0
```

Note that at this point, the outgoing interface information is missing. The CEF switching path is not fully complete until the call is brought up.

On the remote PE, are the necessary tags appearing?

Check that the CEF entry for the remote CE (10.1.17.10) includes the two required tags, one to reach the NAS/PE (its IP address, 10.10.104.12) and another for the remote CE IP address (10.1.17.10) entry in the V1.17.com VRF table:

Example 2-44 show ip cef

```
router# show ip cef vrf V1.17.com 10.1.17.10
10.1.17.10/32, version 163, cached adjacency 10.10.103.113
0 packets, 0 bytes
tag information set
  local tag: VPN-route-head
  fast tag rewrite with Et0/1, 10.10.103.113, tags imposed: {78 82}
via 10.10.104.12, 0 dependencies, recursive
  next hop 10.10.103.113, Ethernet0/1 via 10.10.104.12/32
valid cached adjacency
  tag rewrite with Et0/1, 10.10.103.113, tags imposed: {78 82}
                                                     ^^^^
```

In this example, *78* is the tag to reach 10.10.104.12, the NAS/PE. *82* is the tag for selecting the correct outgoing interface on the VHG/PE.

If the tags do not appear on the remote PE, there is a problem in the MPLS VPN core. Troubleshooting MPLS VPN is beyond the scope of this document. For more information, see the MPLS VPN Troubleshooting Guide.

Step 3. Is CEF Switching Configured to Trigger the Call?

In Step 2 above, the outgoing interface in the show mpls command on the NAS/PE was not yet filled in. When a packet comes in from the remote PE with a tag header 82 – the tag for selecting the outgoing interface – the NAS/PE does not yet know what interface to use.

On the NAS/PE, the incoming packet will be processed by the CEF switching path. In this step, check that CEF switching entries have been defined to trigger the call. In the example, the CEF switching path for 10.1.17.10 (the CE) shows that the IP CEF switching path has a valid CEF entry to reach that address and Dialer50.

This entry makes it possible to trigger the call because the tag 82 in the VRF FIB table shows it is linked to a route for 10.1.17.10, which is in the VRF FIB as an *aggregate*.

With an aggregate route, CEF lookup is not done via the CEF fast switching path and information in the adjacency table, but is handled by the IP CEF process switching routine. Packets are handed off to the dialer routine code.

Note Because the call is not yet established, the CEF entry is still incomplete – tag rewrite is empty and there is no reference to the adjacency for the outgoing interface. When the call is established, this information will be completed.

Example 2-45 show ip cef

```
router# show ip cef vrf V1.17.com 10.1.17.10
10.1.17.10/32, version 9, epoch 0, attached
0 packets, 0 bytes
  tag information set
    local tag: 82
  via Dialer50, 0 dependencies
    valid punt adjacency
    tag rewrite with , , tags imposed: {} (***see note below)
c72d2-2#sh ip cef vrf V1.17.com dialer 50
Prefix              Next Hop             Interface
10.1.17.10/32       attached             Dialer50
```

Part 2. After the Call Is Brought Up

Is the Call Triggered and a PPP Connection Established?

With CEF switching properly configured on the NAS/PE, if a packet comes in on the ingress MPLS interface with a tag header containing tag value 82, the NAS/PE looks at the IP CEF switching path in the VRF and determined that it must go via dialer 50.

At this point the IP packet gets punted to the dialer code, and the dialer routine triggers the call and brings up the PPP connection to the remote CE. As soon the connection is up, the MPLS and CEF switching procedures will adjust to the new changes.

On the NAS/PE, is the route connected?

In this example, the route is connected (the route metric is changed from static to connected):

Example 2-46 show ip route

```
router# show ip route vrf V1.17.com connected
     10.0.0.0/32 is subnetted, 6 subnets
        10.1.17.10 is directly connected, Dialer50
```

On the NAS/PE, is the show mpls information updated?

In this example, the show mpls information is updated for tag 82. Note that the tag relationship has changed from aggregate to untagged, and the outgoing interface uses vi2:

Example 2-47 show mpls forwarding table

```
router# sh mpls forwarding-table vrf V1.17.com 10.1.17.10
Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
82     Untagged    10.1.17.10/32[V]  722072     Vi2        point2point
```

Is the CEF switching path updated?

With the change from aggregate to untagged in the show mpls information above, CEF lookup is no longer done via the IP CEF process switching path.

With an outgoing interface (vi2) stipulated, the CEF lookup process will now look for adjacencies for this outgoing interface. CEF fast switching is in place, as shown in the following example:

Example 2-48 show adjacency

```
router# show adjacency virtual-acc 2
Protocol Interface                 Address
TAG      Virtual-Access2           point2point(4) (incomplete)
IP       Virtual-Access2           point2point(33)
```

Example 2-49 show adjacency detail

```
router# show adjacency virtual-acc 2 detail
Protocol Interface                 Address
TAG      Virtual-Access2           point2point(4) (incomplete)
                                   0 packets, 0 bytes
                                   mpls adj   never
                                   Epoch: 0
IP       Virtual-Access2           point2point(33)
                                   8098 packets, 842192 bytes
                                   FF030021
                                   Epoch: 0
```

The CEF lookup path is changed in order to process the packets faster. Packets are now processed by checking the adjacency entries.

Note Although the TAG path shows incomplete, this is normal since the tag header of the incoming MPLS packet should not be forwarded over the PPP link. Since the TAG path is incomplete, it will now look at the IP adjacency when forwarding the IP header and data received from the MPLS cloud and append the header "FF030021" in front of it.

Debugging Commands for Direct ISDN PE Dial-out

For general dial-out troubleshooting, the following debug commands may be used. Debug commands are issued in enable mode. Debug output examples are shown in Table 2-5.

Table 2-5 Debug Commands for Direct ISDN PE Dial-out

Command	Use to
debug dialer	Show information on the dialer profile.
deb ip cef dialer	Show the change in the CEF switching path, as discussed in Part 2 above.
deb mpls pac	Show information on the mpls packet. **Caution** In a production environment, this command can produce much more output than the packet you are interested in.

Debugging Examples

Example 2-50 Debugging Direct ISDN PE Dial-out

```
c72d2-2#
Mar  7 09:58:37.878: TAG: PO5/0: recvd: CoS=0, TTL=252, Tag(s)=82
This is the incoming MPLS packet with a tag header containing tag value 82, destined for
remote CE and triggering the call. The following debugging shows this.
Mar  7 09:58:37.882: Vi31 DDR: Dialing cause ip (s=10.2.17.241, d=10.1.17.10)
Mar  7 09:58:37.882: Vi31 DDR: Attempting to dial 11710
Mar  7 09:58:37.914: %LINK-3-UPDOWN: Interface Virtual-Access31, changed state to up
Mar  7 09:58:37.914: Vi31 DDR: Dialer statechange to up
Mar  7 09:58:37.914: %DIALER-6-BIND: Interface Vi31 bound to profile Di50
Mar  7 09:58:37.914: Vi31 DDR: Dialer call has been placed
Mar  7 09:58:37.966: %DIALER-6-BIND: Interface Vi30 bound to profile Di50
Mar  7 09:58:37.970: %LINK-3-UPDOWN: Interface Virtual-Access30, changed state to up
Mar  7 09:58:37.970: Vi30 DDR: Dialer statechange to up
Mar  7 09:58:37.970: Vi30 DDR: Dialer call has been placed
```

Example 2-51 Debugging After Changes in the CEF Switching Path

This output is seen only if the call comes up. If the call does not come up, troubleshooting is the same as for a non-MPLS call.

```
Mar  7 09:58:37.978: CEF-Dialer (legacy): add link to 10.1.17.10 via Dialer50 through
Virtual-Access30
Mar  7 09:58:37.978: CEF-Dialer: adjacency added: 0x62F77AC0
Mar  7 09:58:37.978: CEF-Dialer: adjacency found: 0x62F77AC0; fib->count: 1
Mar  7 09:58:37.978: CEF-Dialer: setup loadinfo with 1 paths
Mar  7 09:58:37.978: Vi30 DDR: dialer protocol up
Mar  7 09:58:38.958: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access31,
changed state to up
Mar  7 09:58:38.970: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access30,
changed state to up
When the call goes down
deb ip cef dialer
Mar  7 12:41:47.853: CEF-Dialer (legacy): remove link to 10.1.17.10 via Dialer50 through
Virtual-Access30
```

Troubleshooting Specific Features

The features described here may be used with various dial access methods. This section describes how to troubleshoot the feature itself. For information on troubleshooting the main call flow for the specific access method, see the access method sections.

This section includes:

Verifying and Troubleshooting Multilink PPP (on the VHG/PE or NAS/PE)

Multilink PPP (MLP) is designed to exploit additional bandwidth that may be available between two network devices by bundling together two PPP links, with the same IP address assigned to both. From the user's point of view, this appears to be a single link, but because packets can be transferred on both links, it can operate more efficiently. The multilink bundle is always associated with a virtual access interface.

On the VHG/PE (in L2TP dial-in access) or the NAS/PE (in direct dial-in access), use the **show ppp multilink** command in user EXEC mode to verify the following:

- There are no lost or discarded fragments.
- Multilink bundle settings contain the correct settings received from the virtual template and from the service provider AAA server.
- The bundle displays the correct number of member links (2,4 or 6 member links). In Example 2-52, the bundle displays two links.
- The multilink bundle should be placed in the correct VRF via the VRF settings received from the AAA server

Example 2-52 Sample Show Output for Multilink PPP

```
Virtual-Access9, bundle name is U0006N1P3V1.2@V1.2.com/U0006N1P3V1.2@V1.2.com
  Bundle up for 00:00:37
  0 lost fragments, 0 reordered, 0 unassigned
  0 discarded, 0 lost received, 11/255 load
  0x7E received sequence, 0x0 sent sequence
  Member links: 2 (max not set, min not set)
    Serial1/0:0, since 00:00:37, last rcvd seq 00007C
    Serial1/0:1, since 00:00:05, last rcvd seq 00007D
```

For more on troubleshooting MLP, refer to "Configuring and Troubleshooting Multilink PPP", http://www.cisco.com/warp/public/471/top_issues/access/793_multilink.html.

Verifying and Troubleshooting Multichassis Multilink PPP (MMP)

Multichassis Multilink PPP (MMP) provides the capability for MLP links to terminate at multiple "stacked" network access servers. The network access servers are configured as members of a "stack group" and, to the caller, appear as a single PE server.

With MMP, an organization can provide a single dialup telephone number for a large dialup pool. MMP enhances a network's scalability; an organization can add new routers to their dialup pool to expand capacity.

MMP is accomplished through the use of SGBP (Stack Group Bidding Protocol), which assigns ownership of a call to a master NAS in the stack group through a process of bidding. A call flow follows this general sequence, shown in Figure 2-11:

1. MLP Call 1 is made by user X. NAS A answers the call.

2. NAS A informs its stack group peer network access servers that it has accepted a call from user X on CE router X.

3. All members of the stack group bid for the ownership of the call ("bundle mastership").

4. In this example, SGBP bidding was configured to have the network access server that receives the first call "win". NAS A, therefore, becomes the bundle master for the MLP session and receives the call. As bundle master, NAS A owns all connections with user X.

5. When user X needs more bandwidth, a second MLP call (Call 2) is made to the stack group. In this example, NAS C accepts the call, and informs its stack group peers of the call.

6. As in Step 3, the stack group members bid for ownership of this call.

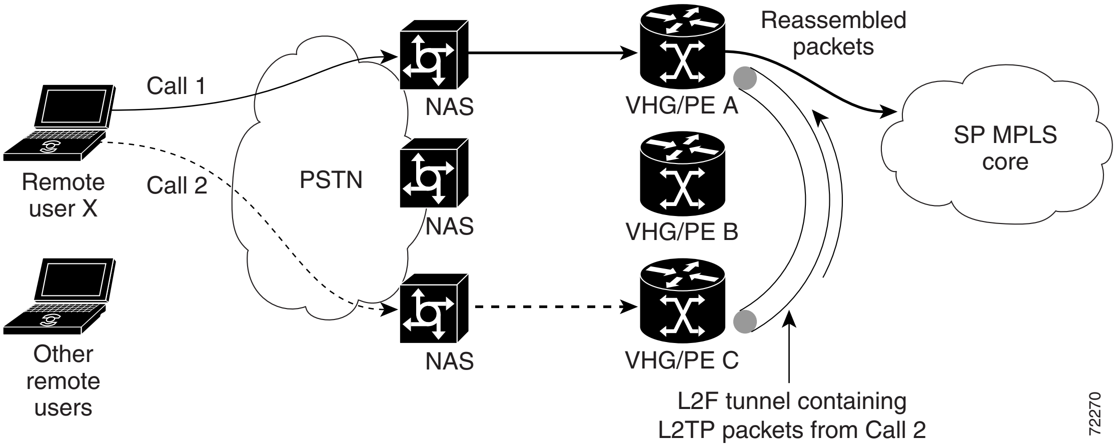
7. NAS A wins the bidding, because it already has an MLP session from user X. NAS C forwards the raw PPP data to NAS A (tunneling via L2F), which reassembles and resequences the packets.

8. Final authentication is done on the bundle master, NAS A.

9. The reassembled data is passed on to the MPLS VPN as if it had all come through one physical link

L2F performs standard PPP operations up to authentication,

Figure 2-11 Topology of Multichassis Multilink PPP



If a problem occurs in multichassis multilink PPP, use the show and debug commands shown in Table 2-6.

For more on troubleshooting MMP, refer to "Multichassis MP, Part 2", http://www.cisco.com/warp/public/131/6.html.

Table 2-6 show and debug Commands for MMP

Command	Use To...
show ppp multilink	Display multilink PPP bundle information.
show sgbp	Verify the state of the stack group members. Make sure all member states are ACTIVE (the group member is functioning as a group member). If state is IDLE, the stack group cannot detect the remote stack group member. Unless there is a known reason for the down state (such as maintenance), look for routing problems. Use **debug sgbp errors** to confirm that authentication was successful. **Note** CONNECTING or WAITINFO are transitional states that should occur only when the stack group member is making the transition to the ACTIVE state.
debug sgbp hellos	Use this command if an authentication problem between two stack group members is detected using **show sgbp**. Stack group names and member names are case-sensitive and must match exactly the host names configured for the stack group. If authentication of one NAS by another fails, determine if a new group member has a password.
debug sgbp error	Check for wiring or routing problems which may cause the stack member's source IP address (received in the hello message) to differ from its locally-defined IP address.
debug vpdn event	Display errors and events associated with establishing or terminating L2TP tunnels for VPDNs.
debug vpdn error	Display errors associated with L2TP events.
debug vpdn l2f-error	Display errors associated with L2X protocol events.
debug vtemplate	Display cloning information for a virtual access interface from the time it is cloned from a virtual template to the time it comes down.

Example 2-53 shows the command output on stack group member router D where user X has its bundle interface as Virtual-Access4. Two member interfaces are joined to this bundle interface:

- The first is a local PRI channel
- The second is a projected interface from stack group member NAS B.

Example 2-53 Sample Results of show ppp multilink for Troubleshooting MMP

```
RouterD# show ppp multilink
```

```
Bundle UserX 2 members, Master link is Virtual-Access4
0 lost fragments, 0 reordered, 0 unassigned, 100/255 load
0 discarded, 0 lost received, sequence 40/66 rcvd/sent
members: 2
  NASC: Virtual-Access1 (1.1.1.3)
  NASB: Virtual-Access6 (1.1.1.2)
```

In Example 2-54, the stack group member NASA cannot detect the remote stack group member router D.

Example 2-54 Sample Results of show sgbp for Troubleshooting MMP

```
NASA# show sgbp

Group Name: stack1 State: 0 Ref: 0xC07B060
Member Name: NASB State: ACTIVE  Id: 1  Ref: 0xC14256F
 Address: 1.1.1.2 Tcb: 0x60B34538
Member Name: NASC State: ACTIVE  Id: 2 Ref: 0xA24256D
 Address: 1.1.1.3 Tcb: 0x60B34439
Member Name: RouterD State: IDLE Id: 3 Ref: 0x0
 Address: 1.1.1.4 Tcb: 0x0
```

In Example 2-55, authentication fails because the new group member (NASA) does not have a password.

Example 2-55 Sample Results of debug sgbp hellos for Troubleshooting MMP

```
NASA# debug sgbp hellos

%SGBP-7-CHALLENGE: Send Hello Challenge to NASB group stack1
%SGBP-7-CHALLENGED: Hello Challenge message from member NASB
using (1.1.1.2)
%SGBP-7-RESPONSE: Send Hello Response to NASB group stack1
%SGBP-7-RESPONDED: Hello Response message from member NASB
using (1.1.1.2)
%SGBP-7-AUTHOK: Send Hello Authentication OK to member NASBusing (1.1.1.2)
%SGBP-7-INFO: Addr = 1.1.1.2 Reference = 0xC347DF7
%SGBP-5-ARRIVING = New peer event for member NASB
%SGBP-7-AUTHFAILED - Member NASB failed authentication
%SGBP-7-NORESP - Fail to respond to NASB group stack1, may not have password
```

The **debug sgbp queries** command displays SGBP bundle mastership queries. In Example 2-56, router D becomes the master for a bundle from user X.

Example 2-56 Sample Results of debug sgbp queries for Troubleshooting MMP

```
RouterD# debug sgbp queries

will show the query state transition
%SGBPQ-7-MQ:  Bundle: UserX   State: Query_to_peers OurBid: 050
%SGBPQ-7-PB:1.1.1.1     State: Open_to_peer    Bid: 000     Retry: 0
%SGBPQ-7-PB:1.1.1.2     State: Open_to_peer    Bid: 000     Retry: 0
%SGBPQ-7-PB:1.1.1.3    State: Open_to_peer    Bid: 000     Retry: 0

%SGBPQ-7-MQ:  Bundle: UserX   State: Query_to_peers OurBid: 050
%SGBPQ-7-PB:1.1.1.1    State: Rcvd  Bid: 000     Retry: 0
%SGBPQ-7-PB:1.1.1.2     State: Rcvd  Bid: 000     Retry: 0
%SGBPQ-7-PB:1.1.1.3     State: Rcvd  Bid: 000     Retry: 0
```

```
%SGBPQ-7-DONE: Query #5 for bundle UserX, count 1, master is local
%SGBPQ-7-MQ:  Bundle: UserX   State: Done     OurBid: 10000
%SGBPQ-7-PB:1.1.1.1      State: Rcvd  Bid: 000      Retry: 0
%SGBPQ-7-PB:1.1.1.2    State: Rcvd     Bid: 000      Retry: 0
%SGBPQ-7-PB:1.1.1.3    State: Rcvd     Bid: 000      Retry: 0
```

In Example 2-57, **debug sgbp error** is used to detect whether the configured access server address is different from the hello address.

In the first **debug sgbp error**, issued from RouterD, the first line shows that the SGBP hello received from NASB does not match the IP address configured (with **sgbp member**) on the local system. To correct the problem, check the configuration of NASB or the configuration of the stack group on the local NAS.

The second line shows that NASK is not a local stack group member.

In the second **debug sgbp error**, issued from NASC, the group member name and the host name are mismatched because of a difference in case: the CHAP challenge is sent with the host name "routerd", the server is actually configured as "RouterD".

Example 2-57 Sample Results of debug sgbp error for Troubleshooting MMP

```
RouterD# debug sgbp error

%SGBP-7-DIFFERENT - NASB addr 1.1.1.2 is different from hellos
addr (3.3.4.5)
%SGBP-7-MISCONF, Possible misconfigured member NASK (1.1.1.6)

NASC# debug sgbp error
%SGBP-7-CHALLENGE: Send Hello Challenge to routerd group stack1
%SGBP-1-MISSCONF: Possible misconfigured member RouterD using 1.1.1.4
```

In this example, **debug sgbp error** is issued on NASC (the network access server side of the CHAP exchange), and shows that the password on RouterD is incorrect.

```
NASC# debug sgbp error

%SGBP-7-CHALLENGED: Rcv Hello Challenge message from member RouterD using 1.1.1.4
%SGBP-7-RESPONSE: Send Hello Response to RouterD group stack1
%SGBP-7-CHALLENGE: Send Hello Challenge to RouterD group stack1
%SGBP-7-RESPONSED: Rcv Hello Response message from member RouterD using 1.1.1.4
%SGBP-1-AUTHFAILED: Member RouterD failed authentication
```

Verifying and Troubleshooting On-demand Address Pools

In on-demand address pools (ODAP), a central SP RADIUS server manages a block of addresses for each customer. Each pool is divided into subnets of various sizes, and the server assigns subnets to the VHG/PE or NAS/PE on request.

The VHG/PE or NAS/PE acts as a DHCP server. On the VHG/PE or NAS/PE, one on-demand pool is configured for each customer VPN supported by that router. Upon configuration, the VHG/PE or NAS/PE's pool manager requests an initial subnet from the server.

Address management is on demand because address pool subnets are allocated or released based on a threshold. If use exceeds a defined ceiling threshold, the pool manager requests an additional subnet from the server and adds it to the on-demand pool. If use falls below a floor threshold, the pool manager attempts to free one, or more then one, of the on-demand pool's subnets to return it to the server. The VRF routing table on the VHG/PE or NAS/PE is updated with the subnet route whenever a range of addresses is requested from the AR.

If a problem occurs in ODAP, use the commands shown in Table 2-7 on the VHG/PE or NAS/PE. Example 2-58 shows the results of **show up dhcp pool** and Example 2-59 shows the results of **debug ip dhcp server events**.

Table 2-7 show and debug Commands for ODAP

Command	Use To...
show ip dhcp pool <address pool name>	Check that DHCP pool hands out IP addresses for incoming PPP session and puts it in the correct VRF.
debug ip dhcp server events	Report server events such as address assignments.

Example 2-58 Sample Results of show ip dhcp pool for Troubleshooting ODAP

```
Router# show ip dhcp pool odap-test
Pool odap-test : Utilization mark (high/low)    : 80 / 20 Subnet size (first/next)       :
27 / 27 (autogrow) VRF name                      : V1.1.com Total addresses
: 30 Leased addresses               : 0 Pending requests               : 0 1 subnet is
currently in the pool :Current index        IP address range            Leased addresses
42.1.1.1             42.1.1.1   - 42.1.1.30
```

Example 2-59 Sample Results of debug ip dhcp server events for Troubleshooting ODAP

```
Router# debug ip dhcp server events
DHCPD: allocate request for client U1000N1P3V1.1@V1.1.com on Virtual-Access7.
DHCPD: locate VRF V1.1.com pool odap-test for client U1000N1P3V1.1@V1.1.com.
DHCPD: assigned IP address 42.1.1.1 to client
5531.3030.304e.3150.3356.312e.3140.5631.2e31.2e63.6f6d.
```

Troubleshooting the Framed-Route VRF Aware Feature

On the VHG/PE, verify that the subnet sent to the CPE is in the appropriate VRF routing table:

show ip route vrf <vrf name>

If the subnet is not in the correct VRF routing table, troubleshoot the RADIUS exchange between the VHG/PE and the RADIUS AR server, checking to make sure the AV pair containing the subnet is being exchanged. Use the following commands:

debug aaa authorization

debug aaa authentication

debug aaa per-user

debug radius

debug ip routing vrf *vrf name to which PPP session belongs*

Example 2-60 Example of VHG/PE show ip route Command Output

```
c72d9-1#
```

```
*Sep  4 09:42:33.627: AAA/AUTHOR (0x55): Pick method list 'default'
*Sep  4 09:42:33.631: AAA/AUTHEN/PPP (00000055): Pick method list 'default'
*Sep  4 09:42:33.631: RADIUS: Pick NAS IP for uid=85 tableid=0 cfg_addr=10.10.104.9 
best_addr=0.0.0.0
*Sep  4 09:42:33.631: RADIUS/ENCODE(00000055): acct_session_id: 146
*Sep  4 09:42:33.631: RADIUS(00000055): sending
*Sep  4 09:42:33.631: RADIUS(00000055): Send to unknown id 21647/157 10.10.100.3:1645, 
Access-Request, len 103
*Sep  4 09:42:33.635: RADIUS:  authenticator 96 9E 2F 52 E4 9E 98 10 - E5 B1 B4 77 F5 F4 
40 63
*Sep  4 09:42:33.635: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
*Sep  4 09:42:33.635: RADIUS:  User-Name           [1]   24  "U0001N1P3V1.9@V1.9.com"
*Sep  4 09:42:33.635: RADIUS:  CHAP-Password       [3]   19  *
*Sep  4 09:42:33.635: RADIUS:  NAS-Port-Type       [61]  6   ISDN                      [2]
*Sep  4 09:42:33.635: RADIUS:  Called-Station-Id   [30]  6   "9111"
*Sep  4 09:42:33.635: RADIUS:  Service-Type        [6]   6   Framed                    [2]
*Sep  4 09:42:33.635: RADIUS:  NAS-IP-Address      [4]   6   10.10.104.9
*Sep  4 09:42:33.635: RADIUS:  Acct-Session-Id     [44]  10  "00000092"
*Sep  4 09:42:33.639: RADIUS: Received from id 21647/157 10.10.100.3:1645, Access-Accept, 
len 478
*Sep  4 09:42:33.639: RADIUS:  authenticator AA 76 9F 6E 15 06 14 5D - 4B DA F0 6C E6 25 
D3 C4
*Sep  4 09:42:33.639: RADIUS:  Service-Type        [6]   6   Framed                    [2]
*Sep  4 09:42:33.639: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
*Sep  4 09:42:33.639: RADIUS:  Vendor, Cisco       [26]  83
*Sep  4 09:42:33.639: RADIUS:   Cisco AVpair       [1]   77  "lcp:interface-config=ip vrf 
forwarding V1.9.com \n ip unnumbered loopback 9"
*Sep  4 09:42:33.639: RADIUS:  Vendor, Cisco       [26]  75
*Sep  4 09:42:33.639: RADIUS:   Cisco AVpair       [1]   69  "ip:route=vrf V1.9.com 
192.168.200.0 255.255.255.0 32.1.9.10 tag 250"
*Sep  4 09:42:33.639: RADIUS(00000055): Received from id 21647/157
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: service-type
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: Framed-Protocol
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: interface-config:Peruser I/F
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: route:Peruser
*Sep  4 09:42:33.663: %LINK-3-UPDOWN: Interface Virtual-Access10, changed state to up
*Sep  4 09:42:33.663: AAA/AUTHEN/PPP (00000055): Pick method list 'default'
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: Process Author
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: Process Attr: interface-config
*Sep  4 09:42:33.663: AAA/AUTHOR: Processing PerUser AV interface-config
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: Process Attr: interface-config
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: IF_config: 
ip vrf forwarding V1.9.com \n ip unnumbered loopback 9

*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: FSM authorization not needed
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/FSM: We can start IPCP
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Start.  Her address 32.1.9.10, we want 0.0.0.0
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Reject 32.1.9.10, using 0.0.0.0
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Processing AV route
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Authorization succeeded
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Done.  Her address 32.1.9.10, we want 0.0.0.0
*Sep  4 09:42:33.727: AAA/AUTHOR: Processing PerUser AV route
*Sep  4 09:42:33.727: Vi10 AAA/PERUSER/ROUTE: route string: IP route vrf V1.9.com 
192.168.200.0 255.255.255.0 32.1.9.10 tag 250

*Sep  4 09:42:33.735: RT(V1.9.com): closer admin distance for 32.1.9.10, flushing 1 routes
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED queued, Queue size 1
*Sep  4 09:42:33.735: RT(V1.9.com): add 32.1.9.10/32 via 0.0.0.0, connected metric [0/0]
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED push
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED queued, Queue size 2
*Sep  4 09:42:33.747: AAA/PER-USER: command = [IP route vrf V1.9.com 192.168.200.0 
255.255.255.0 32.1.9.10 tag 250
]
*Sep  4 09:42:33.747: AAA/PER-USER: line = [IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250]
*Sep  4 09:42:33.751: RT(V1.9.com): add 192.168.200.0/24 via 32.1.9.10, static metric
[1/0]
*Sep  4 09:42:33.751: RT(V1.9.com): NET-RED 192.168.200.0/24
*Sep  4 09:42:33.751: RT(V1.9.com): NET-RED queued, Queue size 1
*Sep  4 09:42:33.763: is_up: 1 state: 4 sub state: 1 line: 0 has_route: True
*Sep  4 09:42:34.663: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access10,
changed state to up
```

When you disconnect, you will see the static route being removed:

```
*Sep  4 09:56:43.713: %LINK-3-UPDOWN: Interface Virtual-Access10, changed state to down
*Sep  4 09:56:43.713: is_up: 0 state: 0 sub state: 1 line: 0 has_route: True
*Sep  4 09:56:43.713: RT(V1.9.com): interface Virtual-Access10 removed from routing table
*Sep  4 09:56:43.713: RT(V1.9.com): Pruning routes for Virtual-Access10 (1)
*Sep  4 09:56:43.713: RT(V1.9.com): delete route to 32.1.9.10 via 0.0.0.0,
Virtual-Access10
*Sep  4 09:56:43.713: RT(V1.9.com): no routes to 32.1.9.10, flushing
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED queued, Queue size 1
*Sep  4 09:56:43.713: RT(V1.9.com): add 32.1.9.10/32 via 0.0.0.0, static metric [1/0]
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED queued, Queue size 2
*Sep  4 09:56:44.713: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access10,
changed state to down
c72d9-1#
c72d9-1#
*Sep  4 09:57:03.712: AAA/PER-USER: command = [no IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250]
*Sep  4 09:57:03.712: AAA/PER-USER: line = [no IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250]
*Sep  4 09:57:03.724: AAA/AUTHOR: decrement ref cnt for ip route 192.168.200.0
255.255.255.0 32.1.9.10 to 0
*Sep  4 09:57:03.724: RT(V1.9.com): del 192.168.200.0 via 32.1.9.10, static metric [1/0]
*Sep  4 09:57:03.724: RT(V1.9.com): delete network route to 192.168.200.0
*Sep  4 09:57:03.724: RT(V1.9.com): NET-RED 192.168.200.0/24
*Sep  4 09:57:03.724: RT(V1.9.com): NET-RED queued, Queue size 1


Show ip route output:

c72d9-1#sh ip rout vrf V1.9.com conn
     32.0.0.0/32 is subnetted, 2 subnets
C       32.1.9.10 is directly connected, Virtual-Access10
C       32.1.9.241 is directly connected, Loopback9

c72d9-1#sh ip route vrf V1.9.com stat
U    192.168.200.0/24 [1/0] via 32.1.9.10

 V1.9.com is the VRf to which the PPP session belongs
 U means  per-user static route ( a route downloaded via AAA)
```

Example 2-61 Example of RADIUS debug Command Output

```
Jun 14 12:27:28.969: AAA/BIND: Bind template "V1.33.com" to uid:139
Jun 14 12:27:28.969: AAA/MLIST Ref count of of mlist 0x639668BC raised to 2
Jun 14 12:27:28.969: AAA/AUTHEN/PPP (0000008B): Pick method list 'method_list_V1_33_com'
Jun 14 12:27:28.969: RADIUS: Pick NAS IP 32.1.33.241 (uid:139) from source config
Jun 14 12:27:28.969: RADIUS/ENCODE(0000008B): acct_session_id: 184
Jun 14 12:27:28.969: RADIUS(0000008B): sending
Jun 14 12:27:28.969: AAA/SG/REF_COUNT refcount of server CF000005 increased to 4
Jun 14 12:27:28.969: RADIUS(0000008B): Send to unknown id 21645/182 10.10.132.4:1645,
Access-Request, len 107
Jun 14 12:27:28.969: RADIUS:  authenticator 55 55 7C 43 89 40 E8 D9 - E5 B1 B4 77 5B E1 63
BB
Jun 14 12:27:28.969: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
Jun 14 12:27:28.969: RADIUS:  User-Name           [1]   26  "U0001N1P3V1.33@V1.33.com"
Jun 14 12:27:28.969: RADIUS:  CHAP-Password       [3]   19  *
Jun 14 12:27:28.969: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
Jun 14 12:27:28.969: RADIUS:  Called-Station-Id   [30]  8   "311033"
Jun 14 12:27:28.969: RADIUS:  Service-Type        [6]   6   Framed                    [2]
Jun 14 12:27:28.969: RADIUS:  NAS-IP-Address      [4]   6   32.1.33.241
Jun 14 12:27:28.969: RADIUS:  Acct-Session-Id     [44]  10  "000000B8"
Jun 14 12:27:28.985: RADIUS: Received from id 21645/182 10.10.132.4:1645, Access-Accept,
len 130
Jun 14 12:27:28.985: RADIUS:  authenticator E0 53 5C BB C0 8E E1 C5 - 0A A9 E8 45 23 96 D6
53
Jun 14 12:27:28.985: RADIUS:  Service-Type        [6]   6   Framed                    [2]
Jun 14 12:27:28.985: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]

Jun 14 12:27:28.985: RADIUS:  Framed-Route        [22]  12  "1.1.1.1/32"

Jun 14 12:27:28.985: RADIUS:  Vendor, Cisco       [26]  86
Jun 14 12:27:28.985: RADIUS:   Cisco AVpair       [1]   80  "lcp:interface-config=ip vrf
forwarding V1.33.com \n  ip unnumbered loopback 33"
Jun 14 12:27:28.985: RADIUS(0000008B): Received from id 21645/182
Jun 14 12:27:28.985: AAA/SG/REF_COUNT decreased ref count of server CF000005 to 3
Jun 14 12:27:28.989: ppp136 PPP/AAA: Check Attr: service-type
Jun 14 12:27:28.989: ppp136 PPP/AAA: Check Attr: Framed-Protocol
Jun 14 12:27:28.989: ppp136 PPP/AAA: Check Attr: route:Peruser
Jun 14 12:27:28.989: ppp136 PPP/AAA: Check Attr: interface-config:Peruser I/F
Jun 14 12:27:28.989: AAA/MLIST Ref count of of mlist 0x600000B  lowered to 1
Jun 14 12:27:28.989: VT:Sending vaccess request, id 0x7200010B
Jun 14 12:27:28.989: VT:Processing vaccess requests, 1 outstanding
Jun 14 12:27:28.989: VT:Create and clone interface, Vt10
Jun 14 12:27:28.989: VT[Vi4]:Reuse interface, recycle queue size 2
Jun 14 12:27:28.989: VT[Vi4]:Cloning a recycled vaccess
Jun 14 12:27:28.989: VT[Vi4]:Processing vaccess response, id 0x7200010B, result success
(1)
Jun 14 12:27:28.989: Vi4: Binding template V1.33.com
Jun 14 12:27:28.989: AAA/BIND: Bind Virtual-Access4 to uid:139 (ccb:0x63EF9738)
Jun 14 12:27:28.993: %LINK-3-UPDOWN: Interface Virtual-Access4, changed state to up
Jun 14 12:27:28.993: Vi4 AAA/AUTHOR/LCP: Process Author
Jun 14 12:27:28.993: Vi4 AAA/AUTHOR/LCP: Process Attr: interface-config
Jun 14 12:27:28.993: AAA/AUTHOR: Processing PerUser AV interface-config
Jun 14 12:27:28.993: Vi4 AAA/AUTHOR/LCP: Process Attr: interface-config
Jun 14 12:27:28.993: Vi4 AAA/AUTHOR/LCP: IF_config:
ip vrf forwarding V1.33.com \n  ip unnumbered loopback 33

Jun 14 12:27:28.993: VT[Vi3]:Reuse interface, recycle queue size 1
Jun 14 12:27:28.993: VT[Vi3]:Vaccess created
Jun 14 12:27:28.993: VT[Vi3]:Created unnumbered vaccess
Jun 14 12:27:28.997: VT[Vi3]:Bringing up interface
Jun 14 12:27:28.997: Vi3 AAA/AUTHOR/LCP: Authorization succeeds trivially
Jun 14 12:27:29.001: %LINK-3-UPDOWN: Interface Virtual-Access3, changed state to up
Jun 14 12:27:29.001: AAA/MLIST Ref count of of mlist 0x62D905E4 raised to 2
Jun 14 12:27:29.001: AAA/MLIST Ref count of of mlist 0x62D905E4 raised to 3
Jun 14 12:27:29.001: Vi3 AAA/AUTHOR/IPCP: FSM authorization not needed
Jun 14 12:27:29.001: Vi3 AAA/AUTHOR/FSM: We can start IPCP
Jun 14 12:27:29.001: RADIUS/ENCODE(0000008B): Unsupported AAA attribute start_time
Jun 14 12:27:29.001: RADIUS/ENCODE(0000008B): Unsupported AAA attribute timezone
Jun 14 12:27:29.001: RADIUS/ENCODE(0000008B): Unsupported AAA attribute start_time
Jun 14 12:27:29.001: RADIUS: Pick NAS IP 32.1.33.241 (uid:139) from source config
Jun 14 12:27:29.001: RADIUS(0000008B): sending
Jun 14 12:27:29.001: Vi3 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 32.1.33.10
Jun 14 12:27:29.001: Vi3 AAA/AUTHOR/IPCP: Processing AV route
Jun 14 12:27:29.001: Vi3 AAA/AUTHOR/IPCP: Authorization succeeded
Jun 14 12:27:29.001: Vi3 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we want 32.1.33.10
Jun 14 12:27:29.001: AAA/SG/REF_COUNT refcount of server CF000005 increased to 4
Jun 14 12:27:29.001: RADIUS(0000008B): Send to unknown id 21645/183 10.10.132.4:1646, 
Accounting-Request, len 203
Jun 14 12:27:29.001: RADIUS:  authenticator C3 19 4C 21 08 76 11 AC - FF F8 DB AE 76 A4 81 
BB
Jun 14 12:27:29.001: RADIUS:  Acct-Session-Id     [44]  10  "000000B8"
Jun 14 12:27:29.001: RADIUS:  Tunnel-Server-Endpoi[67]  14  00:"10.10.104.9"
Jun 14 12:27:29.001: RADIUS:  Tunnel-Client-Endpoi[66]  15  00:"10.10.104.22"
Jun 14 12:27:29.001: RADIUS:  Tunnel-Assignment-Id[82]  8   00:"V1.33"
Jun 14 12:27:29.001: RADIUS:  Acct-Tunnel-Connecti[68]  12  "3990802017"
Jun 14 12:27:29.001: RADIUS:  Tunnel-Client-Auth-I[90]  16  00:"c53d9-1-V1.33"
Jun 14 12:27:29.001: RADIUS:  Tunnel-Server-Auth-I[91]  10  00:"c72d9-1"
Jun 14 12:27:29.001: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
Jun 14 12:27:29.001: RADIUS:  Authentic           [45]  6   RADIUS                    [1]
Jun 14 12:27:29.001: RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
Jun 14 12:27:29.001: RADIUS:  User-Name           [1]   26  "U0001N1P3V1.33@V1.33.com"
Jun 14 12:27:29.001: RADIUS:  Multilink-Session-ID[50]  10  "0000008C"
Jun 14 12:27:29.001: RADIUS:  Acct-Link-Count     [51]  6   1
Jun 14 12:27:29.001: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
Jun 14 12:27:29.001: RADIUS:  Called-Station-Id   [30]  8   "311033"
Jun 14 12:27:29.001: RADIUS:  Service-Type        [6]   6   Framed                    [2]
Jun 14 12:27:29.005: RADIUS:  NAS-IP-Address      [4]   6   32.1.33.241
Jun 14 12:27:29.005: RADIUS:  Event-Timestamp     [55]  6   1024057649
Jun 14 12:27:29.005: RADIUS:  Acct-Delay-Time     [41]  6   0
Jun 14 12:27:29.013: AAA/AUTHOR: Processing PerUser AV route

Jun 14 12:27:29.013: AAA/PER-USER: command = [ip route vrf V1.33.com 1.1.1.1 
255.255.255.255 32.1.33.10
]
Jun 14 12:27:29.013: AAA/PER-USER: line = [ip route vrf V1.33.com 1.1.1.1 255.255.255.255 
32.1.33.10]
Jun 14 12:27:29.017: AAA/AUTHOR: increment ref cnt for ip route 1.1.1.1 255.255.255.255 
32.1.33.10 to 8

Jun 14 12:27:29.041: RADIUS: Received from id 21645/183 10.10.132.4:1646, 
Accounting-response, len 20
Jun 14 12:27:29.041: RADIUS:  authenticator 5A 21 1C F0 FF EE FA FC - 90 A7 5C D5 1F 27 E2 
6F
Jun 14 12:27:29.041: AAA/SG/REF_COUNT decreased ref count of server CF000005 to 3
Jun 14 12:27:29.041: AAA/MLIST Ref count of of mlist 0x6A000010 lowered to 2
Jun 14 12:27:29.993: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access4, 
changed state to up
Jun 14 12:27:30.001: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access3, 
changed state to up
```

CHAPTER 3

Troubleshooting DSL Access to MPLS VPN Integration

Chapter Overview

This chapter contains the following information about integrated Digital Subscriber Line (DSL) access to Multiprotocol Label Switching (MPLS) Virtual Private Network (VPN):

- The following methods of DSL access (network environment architectures):
  - RFC 1483 Routing to MPLS VPN Integration, page 3-2
  - RFC 1483 Routed Bridge Encapsulation to MPLS VPN Integration, page 3-7
  - PPPoX Remote Access to MPLS VPN Integration, page 3-13
  - PPPoX Remote Access SSG to MPLS VPN Integration, page 3-26
  - L2TP to MPLS VPN Integration, page 3-40
- Debugging problems associated with each DSL access method:
  - Debugging Problems Associated with RFC 1483 Routing Integration, page 3-3
  - Debugging Problems Associated with RBE to MPLS VPN Integration, page 3-9
  - Debugging Problems Associated with PPPoX Remote Access to MPLS VPN Integration, page 3-15
  - Debugging Problems Associated with PPPoX Remote Access SSG to MPLS VPN Integration, page 3-28
  - Debugging Problems Associated with DSL L2TP to MPLS VPN Integration, page 3-43
- Verifying the correct configuration for each DSL access method:
  - Verifying Correct Configuration for RFC 1483 Routing to MPLS VPN Integration, page 3-6
  - Verifying Correct Configuration for RBE to MPLS VPN Integration, page 3-11
  - Verifying Correct Configuration for PPPoX Remote Access to MPLS VPN Integration, page 3-24
  - Verifying Correct Configuration for PPPoX Remote Access SSG to MPLS VPN Integration, page 3-38
  - Verifying Correct Configuration for DSL L2TP to MPLS VPN Integration, page 3-51

RFC 1483 Routing to MPLS VPN Integration

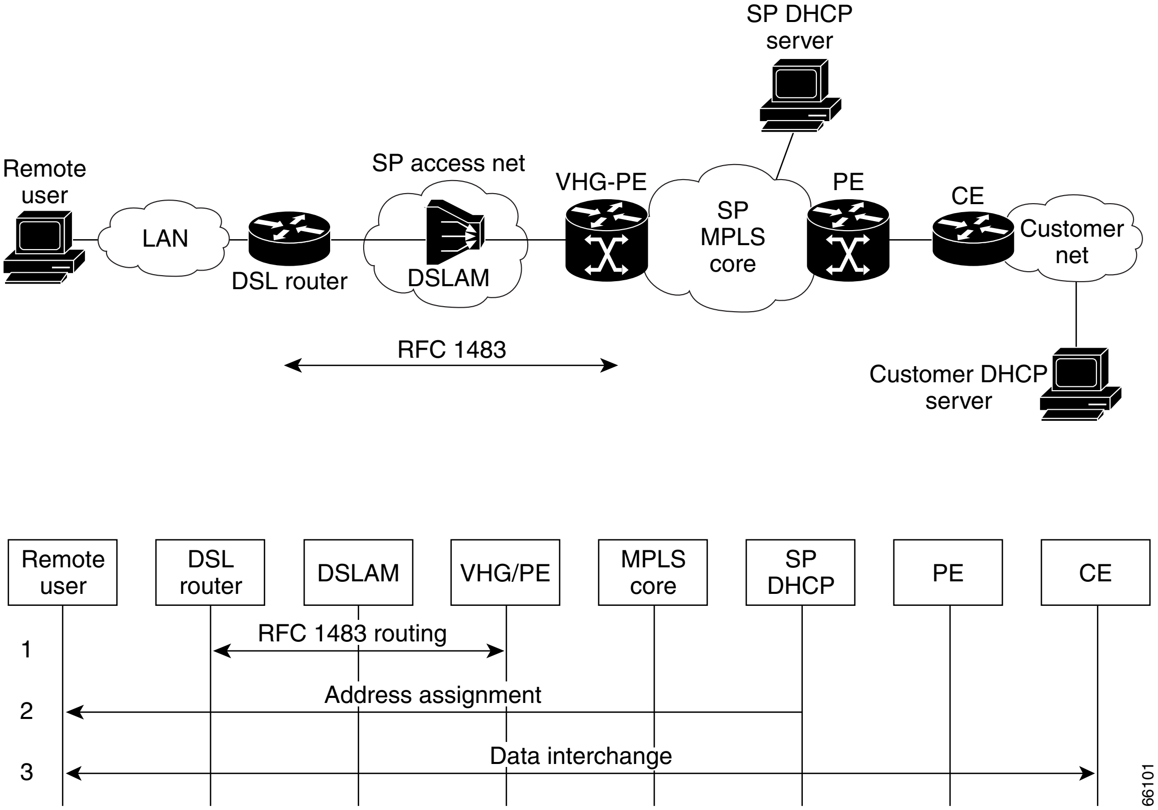
This section contains the following troubleshooting topics:

- Overview of DSL RFC 1483 Routing to MPLS VPN Integration, page 3-2
- Initiating and Viewing debug Command Output, page 3-3
- Debugging Problems Associated with RFC 1483 Routing Integration, page 3-3, including:
  - RFC 1483 PVC Connected (Step 1), page 3-3
  - DHCP Server Provides Address Assignment (Step 2), page 3-5
- Verifying Correct Configuration for RFC 1483 Routing to MPLS VPN Integration, page 3-6

Overview of DSL RFC 1483 Routing to MPLS VPN Integration

Figure 3-1 shows the topology associated with a VPN-capable service provider's MPLS backbone. In this scenario, you should assume that the customer has outsourced all remote access operations to its service provider.

Figure 3-1 DSL RFC 1483 Routing



Initiating and Viewing debug Command Output

For reminders on using the command line interface for viewing debug output, refer to "Initiating and Viewing Command Output" section on page 1-2.

Debugging Problems Associated with RFC 1483 Routing Integration

Table 3-1 corresponds to Figure 3-1, providing a cross-reference to the debugging topics contained in this section. The number shown in column one of this table corresponds to the element number shown in the call flow part of Figure 3-1.

Table 3-1 Troubleshooting Cross-References for Figure 3-1

Element Number Shown in Figure 3-1:	Related Troubleshooting Topic(s):
Line 1: RFC 1483 permanent virtual circuit (PVC) connected between DSL router and provider edge (PE) device.	RFC 1483 PVC Connected (Step 1), page 3-3
Line 2: (Depends on setup) Dynamic Host Configuration Protocol (DHCP) server provides address assignment. **Note** The address assignment applies only if the service provider intends to relay DHCP requests from the customer premises equipment (CPE) to a far-end DHCP server. This line is not necessary if an alternative address assignment mechanism is used, such as a statically assigned address block for each CPE.	DHCP Server Provides Address Assignment (Step 2), page 3-5

RFC 1483 PVC Connected (Step 1)

If RFC 1483 PVC does not connect, use the following **debug** commands in privileged EXEC mode:

- **debug atm packet**—Displays all process-level ATM packets for both outbound and inbound packets.
- **debug ip packet**—Displays general IP debugging information and IP security option (IPSO) security transactions.

The **debug atm packet** command is used to ensure that you are receiving RFC 1483 routed protocol data units (PDUs). An RFC 1483 routed PDU has an 8 byte header. The first 3 bytes are considered the Logical Link Control (LLC) portion [0xAA-AA-03]. The next 3 bytes are the Organizationally Unique Identifier (OUI) portion [0x00-00-00]. The last 2 bytes are the EtherType [0x08-00]. The packet payload follows the header. Example 3-1 shows an example of an RFC 1483 routed PDU and Example 3-2 shows, for comparison, the format of a bridged (802.3-specific) RFC 1483 PDU. Cisco IOS typically does not transmit (but is able to receive) the LAN FCS, CBOS does transmit the LAN FCS, and certain vendors include the LAN FCS.

Example 3-1 RFC 1483 Routed PDU

```
LLC (bytes 1 - 3) - [0xAA-AA-03]
OUI (bytes 4 - 6) - [0x00-00-00]
```

```
EtherType (bytes 7 - 8) - [0x08-00]
payload
```

Example 3-2 Bridged (802.3-Specific) RFC 1483 PDU

```
LLC (bytes 1 - 3) - [0xAA-AA-03]
OUI (bytes 4 - 6) - [0x00-08-C2]
PID (bytes 7 - 8) - [0x00-07] or [0x00-01] if LAN FCS is needed / included
PAD (bytes 9-10) - [0x00 - 00]
Standard MAC header
Payload
LAN FCS (optional)
```

Example 3-3 provides a sample of the output that results from implementing the **debug atm packet** command.

Example 3-3 Sample Debug RFC 1483 PVC Connected

```
nrp1bot# debug atm packet

ATM packets debugging is on
Displaying all ATM packets

nrp1bot# sh debug

Generic ATM: ATM packets debugging is on

[PE]
May  3 16:29:34.713: ATM0/0/0.135(I):
VCD:0x5 VPI:0xA VCI:0x23 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800
Length:0x28
May  3 16:29:34.713: 4500 001C 001F 0000 3F01 65B7 0A01 0101 0A01 0109 0800
F7EC 0000 0013
May  3 16:29:34.713:
```

Example 3-4 shows sample output from the **debug ip** command. Always combine the **debug ip packet detail** command with an access list to prevent overloading the router.

Example 3-4 Sample debug ip Command Output

```
nrp1bot# debug ip packet

IP packet debugging is on

nrp1bot# sh debug

Generic IP: IP packet debugging is on

[PE]

DSL7200# ping 209.165.200.224

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.224, timeout is 2 seconds:
*Oct 4 20:47:57.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, sending
*Oct 4 20:47:57.781: ICMP type=8, code=0
*Oct 4 20:47:57.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, encapsulation
failed
*Oct 4 20:47:57.785: ICMP type=8, code=0.
*Oct 4 20:47:59.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, sending
*Oct 4 20:47:59.781: ICMP type=8, code=0
```

```
*Oct 4 20:47:59.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, encapsulation
failed
```

If a Layer 3 address (for example an IP address) cannot be mapped to a corresponding Layer 2 address, Cisco IOS marks an encapsulation failure on the packet and drops it. So on a multipoint interface, a static map is needed to the next hop or a dynamic map (learned through an ATM Inverse Address Resolution Protocol transaction) to forward packets.

In the case of a static map, you can confirm the configuration of the map statements with a **show atm map** command.

Note IP packet debugging and Asynchronous Transfer Mode (ATM) packet debugging only show packets and cells that are process switched. On MPLS-enabledlabel edge router (LER) and label switch router (LSR) routers, Cisco Express Forwarding (CEF) is configured globally. Hence to view the output for all packets transiting an interface, you can configure the **no ip route-cache cef** command on the ATM interface to display the debug output.

Note Remember to reenable the **ip route-cache cef** command on the interface after completing your debugging.

DHCP Server Provides Address Assignment (Step 2)

If address assignment fails, use the following **debug** commands in privileged EXEC mode:

- **debug ip dhcp server events**—Reports server events, like address assignments and database updates.
- **debug ip dhcp server packet**—Decodes DHCP receptions and transmissions.

Example 3-5 provides a sample of the **debug** command output that results from these commands.

Example 3-5 Sample Debug Address Assignment

```
nrp1bot# sh debug

DHCP server packet debugging is on.
DHCP server event debugging is on.
nrp1bot#
nrp1bot#
May 18 16:43:02.485: DHCPD: setting giaddr to 10.1.1.5.
May 18 16:43:02.485: DHCPD: BOOTREQUEST from 0100.10a4.facc.aa forwarded to 172.29.51.239.
May 18 16:43:02.489: DHCPD: BOOTREQUEST from 0100.10a4.facc.aa forwarded to 172.29.51.239.
May 18 16:43:02.561: DHCPD: setting giaddr to 10.1.1.5.
May 18 16:43:02.561: DHCPD: BOOTREQUEST from 0100.10a4.facc.aa forwarded to 172.29.51.239.
May 18 16:43:02.565: DHCPD: BOOTREQUEST from 0100.10a4.facc.aa forwarded to 172.29.51.239.
May 18 16:43:02.665: hardware_address(): hardware address specified
May 18 16:43:02.665:  hardware_address(): chaddr in pak. = 616874D8
May 18 16:43:02.665: DHCPD: forwarding BOOTREPLY to client 0010.a4fa.ccaa.
May 18 16:43:02.665: DHCPD: Forwarding reply on numbered intf
May 18 16:43:02.665: DHCPD: creating ARP entry (10.1.1.2, 0010.a4fa.ccaa).
May 18 16:43:02.665: DHCPD: SIOCSARP ioctl failed (error 22).
May 18 16:43:02.665: DHCPD: broadcasting BOOTREPLY to client 0010.a4fa.ccaa.
nrp1bot#
```

Verifying Correct Configuration for RFC 1483 Routing to MPLS VPN Integration

The following **show** commands are useful in debugging RFC 1483:

- **show ip route vrf** *vpn*—Displays the IP routing table associated with a VRF (VPN routing/forwarding instance).
- **show interfaces atm** *interface*—Displays information about the ATM interface.
- **show atm map**—Displays the list of all configured ATM static maps to remote hosts on an ATM network and on ATM bundle maps.
- **show atm vc**—Displays all ATM PVCs and switched virtual circuits (SVCs) and traffic information.

The following examples show the output from these **show** commands.

Example 3-6 Sample show ip route vrf <vpn> Command Output

```
nrp1bot# sh ip route vrf vpn100

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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     172.29.0.0/24 is subnetted, 1 subnets
B       172.29.51.0 [200/0] via 10.1.1.4, 7w0d
     10.0.0.0/32 is subnetted, 2 subnets
C       10.1.1.10 is directly connected, Loopback100
B       10.1.1.9 [200/0] via 10.1.1.4, 7w0d
```

Example 3-7 Sample show interfaces atm <interface> Command Output

```
ATM0/0/0.2033 is up, line protocol is up
  Hardware is ATM-SAR
  Description: 1cardPPPoA [20/33]8/0/1->6260->677[1/1]
  MTU 4470 bytes, BW 156250 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM
  100254 packets input, 1535132 bytes
  1313957 packets output,22087983 bytes
  0 OAM cells input, 0 OAM cells output

nrp1bot# sh int atm 0/0/0

ATM0/0/0 is up, line protocol is up
  Hardware is ATM-SAR
  MTU 4470 bytes, sub MTU 4470, BW 156250 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not supported
  Keepalive not supported
  Encapsulation(s): AAL5, PVC mode
  2047 maximum active VCs, 2 current VCCs
  VC idle disconnect time: 300 seconds
  Last input 9w5d, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/80, 7 drops; input queue 0/1000, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     56031 packets input, 1943197 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
     1315498 packets output, 27477357 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Example 3-8 Sample show atm map Command Output

```
DSL7200# show atm map

Map list ATM2/0.7pvc15 : PERMANENT
ip 6.6.6.1 maps to VC 15, VPI 3, VCI 32, ATM2/0.7 , broadcast
```

Example 3-9 Sample show atm vc Command Output

```
nrp1bot# sh atm vc

            VCD /                                      Peak  Avg/Min Burst
Interface   Name       VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells Sts
0/0/0.133  3           10    33   PVC    SNAP     UBR       0    INAC
0/0/0.134  4           10    34   PVC    SNAP     UBR       0    INAC
0/0/0.135  5           10    35   PVC    SNAP     UBR  155000    UP
```

RFC 1483 Routed Bridge Encapsulation to MPLS VPN Integration

This section contains the following troubleshooting topics:

- Overview of RBE to MPLS VPN Integration, page 3-7
- Initiating and Viewing debug Command Output, page 3-8
- Debugging Problems Associated with RBE to MPLS VPN Integration, page 3-9, including:
  - PVC Connected (Step 1), page 3-9
  - DHCP Server Provides Address Assignment (Step 2), page 3-11
- Verifying Correct Configuration for RBE to MPLS VPN Integration, page 3-11

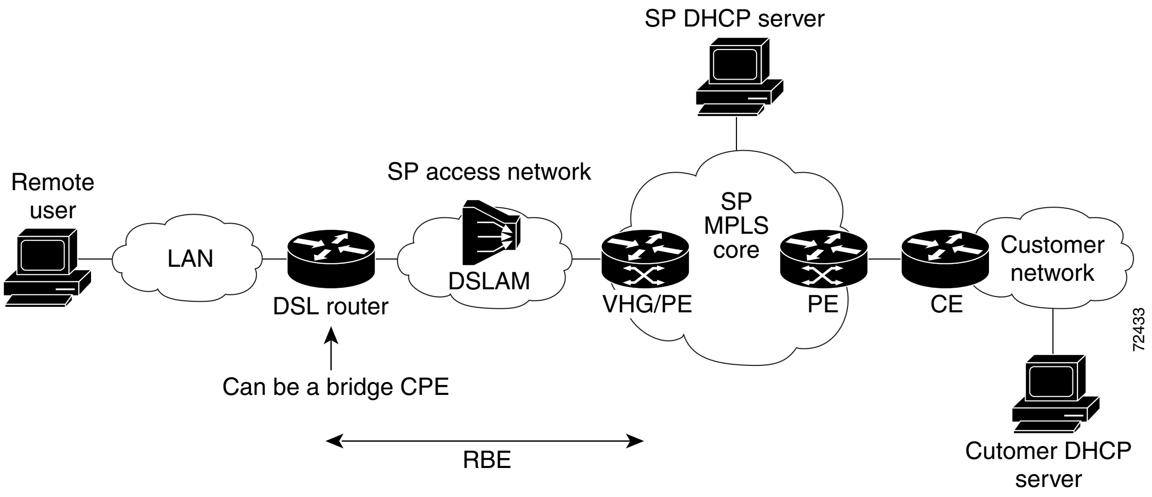
Overview of RBE to MPLS VPN Integration

Routed Bridge Encapsulation (RBE) is used to route RFC1483 ATM Bridged PDUs. The Bridged PDUs are routed by examining the destination IP address within the IP Payload contained within the bridged ethernet frame from a stub-bridged LAN. Bridged IP packets received on an ATM interface configured in routed-bridge mode are routed via an IP header. The interface takes advantage of the characteristics of a stub LAN topology commonly used for DSL access and offers increased performance and flexibility over integrated routing and bridging (IRB).

In Figure 3-2, RBE is configured between the DSL router and the VHG/PE. The DSL router can be set up as a pure bridge or can be set up for IRB, where multiple LAN interfaces are bridged through the bridge group virtual interface (BVI). Each of the DSL routers terminates on a separate point-to-point subinterface on the VHG/PE which is statically configured with a specific VRF. Remote user authentication or authorization is available with Option 82 for DSL routed bridge encapsulation remote access.

RBE treats the VHG/PE subinterface as if it were connected to an Ethernet LAN, but avoids the disadvantages of pure bridging such as broadcast storms, IP hijacking, and ARP spoofing issues. Address management options include static and VRF-aware DHCP servers. Since this architecture is not PPP based, RADIUS accounting cannot be used. Netflow is used for accounting.

Figure 3-2 Cisco VPN DSL RBE to MPLS Integration



RBE References

For a description of RBE architecture, refer to: http://www.cisco.com/warp/public/794/routed_bridged_encap.html.

For RBE IOS commands, refer to http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122sup/122csum/csum2/122cswan/wsfbrda.htm#1051874

For platform-specific overview and configuration information, refer to:

ATM Routed Bridge Encapsulation Feature Overview - Cisco 6400 series:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120limit/120dc/120dc5/atm_rb.htm

ATM Routed Bridge Encapsulation Feature Overview - Cisco 7200 series:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t2/dtatmrbe.htm

Initiating and Viewing debug Command Output

For reminders on using the command line interface for viewing debug output, refer to "Initiating and Viewing Command Output" section on page 1-2.

Debugging Problems Associated with RBE to MPLS VPN Integration

Table 3-2 corresponds to Figure 3-2, providing a cross-reference to the debugging topics contained in this section.

Table 3-2 Troubleshooting RFC 1483 RBE to MPLS VPN Call Flow

Call Flow Step	Related Troubleshooting Topic(s):
Step 1. RFC 1483 permanent virtual circuit (PVC) connected between DSL router and provider edge (PE) device.	RFC 1483 PVC Connected (Step 1), page 3-3
Step 2. (Depends on setup) Dynamic Host Configuration Protocol (DHCP) server provides address assignment. **Note** The address assignment applies only if the service provider intends to relay DHCP requests from the customer premises equipment (CPE) to a far-end DHCP server. This line is not necessary if an alternative address assignment mechanism is used, such as a statically assigned address block for each CPE.	DHCP Server Provides Address Assignment (Step 2), page 3-5

PVC Connected (Step 1)

If RFC 1483 PVC does not connect, use the following **debug** commands in privileged EXEC mode:

- **debug atm packet**—Displays all process-level ATM packets for both outbound and inbound packets.
- **debug ip packet**—Displays general IP debugging information and IP security option (IPSO) security transactions.

The **debug atm packet** command is used to ensure that you are receiving RFC 1483 routed protocol data units (PDUs).

An RFC 1483 routed PDU has an 8 byte header. The first 3 bytes are considered the Logical Link Control (LLC) portion [0xAA-AA-03]. The next 3 bytes are the Organizationally Unique Identifier (OUI) portion [0x00-08-C2]. The type of the bridged media is specified by the two octet PID [0x00-07] or [0x00-010. The PID indicates whether the original Frame Check Sequence (FCS) is preserved within the bridged PDU. Padding is added after the PID field to align the user information field of the bridged PDU at a four octet boundary [0x00 - 00]. The packet payload follows the header. Cisco IOS typically does not transmit (but is able to receive) the LAN FCS, CBOS does transmit the LAN FCS, and certain vendors include the LAN FCS.

Example 3-10 Bridged (802.3-Specific) RFC 1483 PDU

```
LLC (bytes 1 - 3) - [0xAA-AA-03]
OUI (bytes 4 - 6) - [0x00-08-C2]
PID (bytes 7 - 8) - [0x00-07] or [0x00-01] if LAN FCS is needed / included
PAD (bytes 9-10) - [0x00 - 00]
Standard MAC header
Payload
LAN FCS (optional)
```

Example 3-11 provides a sample of the output that results from implementing the **debug atm packet** command.

Example 3-11 Sample Debug RFC 1483 PVC Connected

```
nrp1bot# debug atm packet

ATM packets debugging is on
Displaying all ATM packets

nrp1bot# sh debug

Generic ATM: ATM packets debugging is on

[PE]
May  3 16:29:34.713: ATM0/0/0.135(I):
VCD:0x5 VPI:0xA VCI:0x23 Type:0x0 SAP:AAAA CTL:03 OUI:000000 TYPE:0800
Length:0x28
May  3 16:29:34.713: 4500 001C 001F 0000 3F01 65B7 0A01 0101 0A01 0109 0800
F7EC 0000 0013
May  3 16:29:34.713:
```

Example 3-12 shows sample output from the **debug ip** command. Always combine the **debug ip packet detail** command with an access list to prevent overloading the router.

Example 3-12 Sample debug ip Command Output

```
nrp1bot# debug ip packet

IP packet debugging is on

nrp1bot# sh debug

Generic IP: IP packet debugging is on

[PE]

DSL7200# ping 209.165.200.224

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.224, timeout is 2 seconds:
*Oct 4 20:47:57.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, sending
*Oct 4 20:47:57.781: ICMP type=8, code=0
*Oct 4 20:47:57.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, encapsulation
failed
*Oct 4 20:47:57.785: ICMP type=8, code=0.
*Oct 4 20:47:59.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, sending
*Oct 4 20:47:59.781: ICMP type=8, code=0
*Oct 4 20:47:59.781: IP: s=6.6.6.5 (local), d=6.6.6.6 (ATM2/0.8), len 100, encapsulation
failed
```

If a Layer 3 address (for example an IP address) cannot be mapped to a corresponding Layer 2 address, Cisco IOS marks an encapsulation failure on the packet and drops it. So on a multipoint interface, a static map is needed to the next hop or a dynamic map (learned through an ATM Inverse Address Resolution Protocol transaction) to forward packets.

In the case of a static map, you can confirm the configuration of the map statements with a **show atm map** command.

Note IP packet debugging and Asynchronous Transfer Mode (ATM) packet debugging only show packets and cells that are process switched. On MPLS-enabledlabel edge router (LER) and label switch router (LSR) routers, Cisco Express Forwarding (CEF) is configured globally. Hence to view the output for all packets transiting an interface, you can configure the **no ip route-cache cef** command on the ATM interface to display the debug output.

Note Remember to reenable the **ip route-cache cef** command on the interface after completing your debugging.

DHCP Server Provides Address Assignment (Step 2)

If address assignment fails, use the following **debug** commands in privileged EXEC mode:

- **debug ip dhcp**—Displays debugging information about DHCP client activities and the status of DHCP packets. The **debug dhcp detail** command provides useful information, such as the lease entry structure of the client and the state transitions of the lease entry. The debug output shows the scanned option values from received DHCP messages that are replies to a router request. The values of the op, htype, hlen, hops, server identifier option, xid, secs, flags, ciaddr, yiaddr, siaddr, and giaddr fields of the DHCP packet are shown in addition to the length of the options field.

Example 3-13 provides a sample of the **debug** command output that results from this command.

Example 3-13 Sample Debug Address Assignment

```
nrp1bot# sh debug

DHCPD: relay binding created for client 0100.10a4.f9c2.53.
DHCPD: setting giaddr to 25.0.13.29.
DHCPD: adding relay information option.
DHCPD: BOOTREQUEST from 0100.10a4.f9c2.53 forwarded to 10.15.61.63.
DHCPD: Giaddr from server-id-override suboption 25.0.13.29
DHCPD: forwarding BOOTREPLY to client 0010.a4f9.c253.
        outbound IF index  = 4
        outbound IF sub-index = 1
DHCPD: unnum: broadcasting BOOTREPLY to client 0010.a4f9.c253.

*********************************************************************
RT(red): add 13.0.0.2/32 via 0.0.0.0, static metric [1/0]
        outbound IF index  = 4
        outbound IF sub-index = 1
*********************************************************************rbe
```

Verifying Correct Configuration for RBE to MPLS VPN Integration

The following **show** commands are useful in debugging RBE to MPLS VPN integration:

- **show ip route vrf** *vpn*—Displays the IP routing table associated with a VRF (VPN routing/forwarding instance).
- **show ip arp vrf** *vpn*—Displays the Address Resolution Protocol (ARP) cache associated with a VRF (VPN routing/forwarding instance).
- **show interfaces atm** *interface*—Displays information about the ATM interface.

- **show atm vc**—Displays all ATM PVCs and switched virtual circuits (SVCs) and traffic information.

The following examples show the output from these **show** commands.

Example 3-14 Sample show ip route vrf <vpn> Command Output

```
nrp1bot# sh ip route vrf red
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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     100.0.0.0/32 is subnetted, 3 subnets
B       100.0.0.1 [200/0] via 25.0.13.23, 04:44:23
B       100.0.0.2 [200/0] via 25.0.13.23, 04:44:23
B       100.0.0.3 [200/0] via 25.0.13.23, 04:44:23
     10.0.0.0/30 is subnetted, 1 subnets
B       10.10.10.0 [200/0] via 25.0.13.23, 04:44:23

     13.0.0.0/32 is subnetted, 1 subnets
S       13.0.0.2 is directly connected, ATM4/0.1

nrp1bot# sh ip arp vrf red
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  13.0.0.2                1   0010.a4f9.c253  ARPA   ATM4/0.1
```

Example 3-15 Sample show interfaces atm <interface> Command Output

```
ATM0/0/0.2033 is up, line protocol is up
  Hardware is ATM-SAR
  Description: 1cardPPPoA [20/33]8/0/1->6260->677[1/1]
  MTU 4470 bytes, BW 156250 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM
  100254 packets input, 1535132 bytes
  1313957 packets output,22087983 bytes
  0 OAM cells input, 0 OAM cells output

nrp1bot# sh int atm 0/0/0

ATM0/0/0 is up, line protocol is up
  Hardware is ATM-SAR
  MTU 4470 bytes, sub MTU 4470, BW 156250 Kbit, DLY 80 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ATM, loopback not supported
  Keepalive not supported
  Encapsulation(s): AAL5, PVC mode
  2047 maximum active VCs, 2 current VCCs
  VC idle disconnect time: 300 seconds
  Last input 9w5d, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Queueing strategy: fifo
  Output queue 0/80, 7 drops; input queue 0/1000, 0 drops
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     56031 packets input, 1943197 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
     1315498 packets output, 27477357 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 output buffer failures, 0 output buffers swapped out
```

Example 3-16 Sample show atm vc Command Output

```
nrp1bot# sh atm vc

            VCD /                                      Peak  Avg/Min Burst
Interface   Name       VPI   VCI  Type   Encaps   SC   Kbps   Kbps   Cells Sts
0/0/0.133  3           10    33   PVC    SNAP     UBR       0    INAC
0/0/0.134  4           10    34   PVC    SNAP     UBR       0    INAC
0/0/0.135  5           10    35   PVC    SNAP     UBR  155000    UP
```

PPPoX Remote Access to MPLS VPN Integration

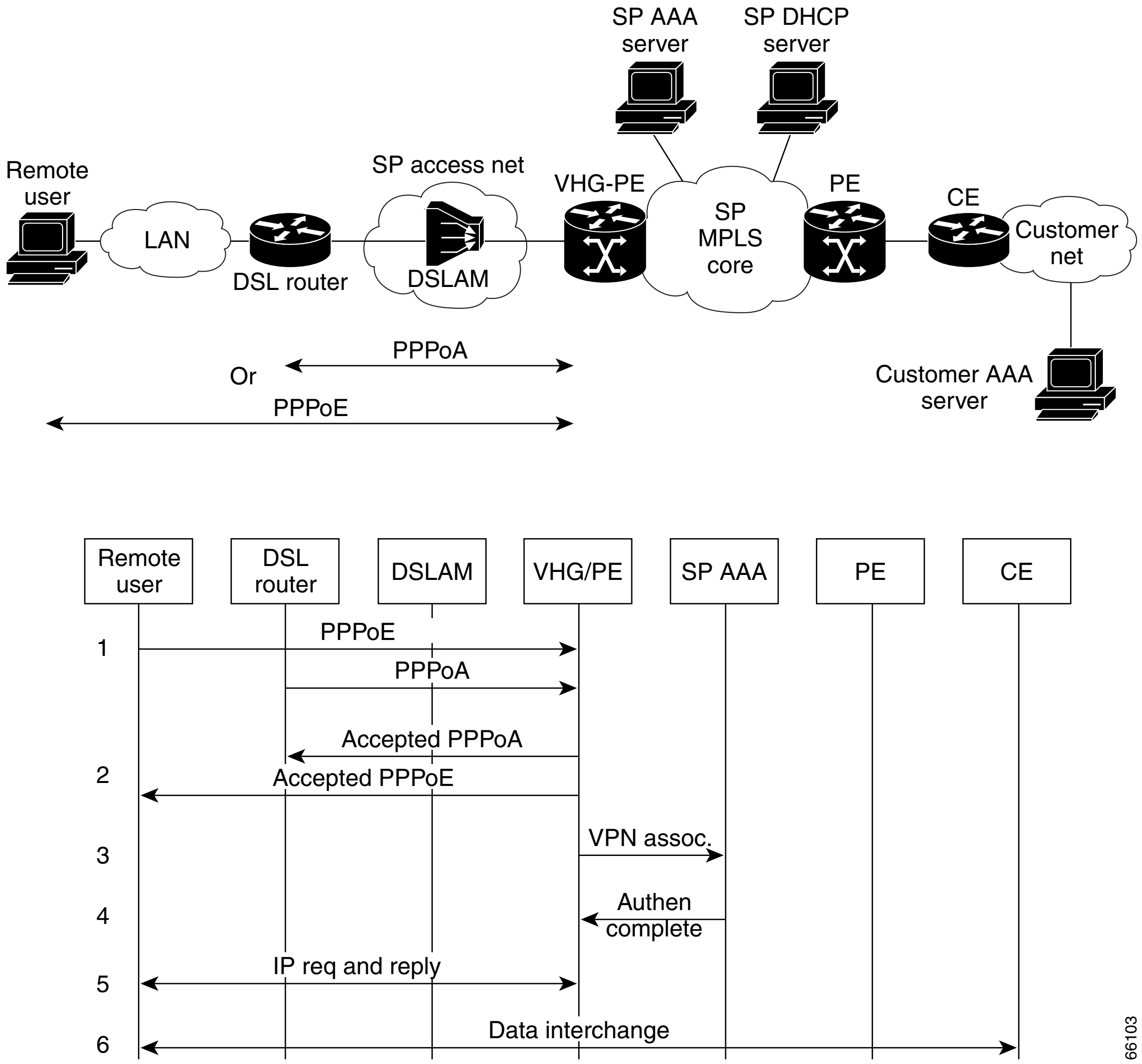
This section contains the following troubleshooting topics:

- Overview of DSL PPPOX Remote Access to MPLS VPN Integration, page 3-13
- Initiating and Viewing debug command Output, page 3-14
- Debugging Problems Associated with PPPoX Remote Access to MPLS VPN Integration, page 3-15, including:
  - DSL Router Initiates a PPPoA Session (Step 1 and Step 2), page 3-15
  - Remote User Initiates a PPPoE Session (Step 1 and Step 2), page 3-17
  - VHG/PE Queries SP RADIUS Server (Step 3 and Step 4[a]), page 3-19
  - Remote User Is Associated with a Specific Customer MPLS VPN (Step 3 and Step 4[b]), page 3-22
  - VHG/PE Obtains IP Address (Step 5), page 3-23
- Verifying Correct Configuration for PPPoX Remote Access to MPLS VPN Integration, page 3-24

Overview of DSL PPPOX Remote Access to MPLS VPN Integration

Figure 3-3 shows the topology associated with a VPN-capable service provider's MPLS backbone. In this scenario, you should assume that the customer has outsourced all remote access operations to its service provider.

Figure 3-3 DSL PPPoX



Initiating and Viewing debug command Output

For reminders on using the command-line interface for viewing debug command output, refer to the sections of the *Remote Access to MPLS VPN Solution Provisioning Guide 1.0* entitled "User Interface Command Modes" and "User Command Modes." This document can be accessed at the following URL:

http://cco/univercd/cc/td/doc/product/vpn/solution/rampls/index.htm

Debugging Problems Associated with PPPoX Remote Access to MPLS VPN Integration

Table 3-3 (below) corresponds to Figure 3-3, providing a cross-reference to the troubleshooting topics associated with the events that occur when the remote user creates a Point-to-Point Protocol over ATM or Ethernet (PPPoX) session over DSL in an attempt to access its corporate network or Internet service provider (ISP).

Table 3-3 Troubleshooting Topics for DSL PPPOX to MPLS VPN Integration

Event Shown in Figure 3-2:	Related Troubleshooting Topic(s):
Lines1-2: DSL router initiates a PPPoA session or remote user initiates a PPPoE session, and VHG/PE accepts session.	DSL Router Initiates a PPPoA Session (Step 1 and Step 2), page 3-15 Remote User Initiates a PPPoE Session (Step 1 and Step 2), page 3-17
Lines 3-4: VHG/PE queries SP RADIUS server and completes the remote user's authentication via RADIUS.	VHG/PE Queries SP RADIUS Server (Step 3 and Step 4[a]), page 3-19
Lines 3-4: After successfully completing authentication, the remote user is associated with a specific customer MPLS VPN.	Remote User Is Associated with a Specific Customer MPLS VPN (Step 3 and Step 4[b]), page 3-22
Line 5: VHG/PE obtains an IP address for the remote user.	VHG/PE Obtains IP Address (Step 5), page 3-23

DSL Router Initiates a PPPoA Session (Step 1 and Step 2)

The DSL router initiates a Point-to-Point Protocol over ATM (PPPoA) session over the DSL access network. If the DSL router cannot initiate a PPPoA session, use the following **debug** commands in privileged EXEC mode:

- **debug ppp negotiation**—Displays PPP packets transmitted during PPP startup, where PPP options are negotiated.
- **debug ppp authentication**—Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.

Example 3-17 provides a sample of the **debug** command output that results from these commands.

Example 3-17 Sample Debug DSL Router Cannot Initiate PPPoA Session

```
nrp1bot# debug ppp negotiation

PPP protocol negotiation debugging is on

nrp1bot# debug ppp authentication

PPP authentication debugging is on

nrp1bot# sh debug

PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
nrp1bot#
Jul  2 15:16:01.502: Vi1 LCP: TIMEout: State Listen
Jul  2 15:16:01.502: Vi1 LCP: O CONFREQ [Listen] id 166 len 15
Jul  2 15:16:01.502: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:01.502: Vi1 LCP:    MagicNumber 0x121E1573 (0x0506121E1573)
Jul  2 15:16:01.534: Vi1 LCP: I CONFACK [REQsent] id 166 len 15
Jul  2 15:16:01.534: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:01.534: Vi1 LCP:    MagicNumber 0x121E1573 (0x0506121E1573)
Jul  2 15:16:02.410: Vi1 LCP: I CONFREQ [ACKrcvd] id 86 len 14
Jul  2 15:16:02.410: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul  2 15:16:02.410: Vi1 LCP:    MRU 2048 (0x01040800)
Jul  2 15:16:02.410: Vi1 LCP: O CONFACK [ACKrcvd] id 86 len 14
Jul  2 15:16:02.410: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul  2 15:16:02.410: Vi1 LCP:    MRU 2048 (0x01040800)
Jul  2 15:16:02.410: Vi1 LCP: State is Open
Jul  2 15:16:02.410: Vi1 PPP: Phase is AUTHENTICATING, by this end [0 sess,
0 load]
Jul  2 15:16:02.410: Vi1 CHAP: O CHALLENGE id 158 len 28 from "nrp1bot"
Jul  2 15:16:02.446: Vi1 CHAP: I RESPONSE id 158 len 34 from "1cardpppoa1_1"
Jul  2 15:16:02.458: Vi1 CHAP: O SUCCESS id 158 len 4
Jul  2 15:16:02.462: Vi1 PPP: Idle timeout, dropping connection
Jul  2 15:16:02.462: Vi1 PPP: Phase is TERMINATING [0 sess, 0 load]
Jul  2 15:16:02.466: Vi1 LCP: O TERMREQ [Open] id 167 len 4
Jul  2 15:16:02.494: Vi1 IPCP: LCP not open, discarding packet
Jul  2 15:16:02.498: Vi1 LCP: I TERMACK [TERMsent] id 167 len 4
Jul  2 15:16:02.498: Vi1 LCP: State is Closed
Jul  2 15:16:02.498: Vi1 PPP: Phase is DOWN [0 sess, 0 load]
Jul  2 15:16:02.506: Vi1 PPP: Phase is ESTABLISHING, Passive Open [0 sess, 0
load]
Jul  2 15:16:02.506: Vi1 LCP: State is Listen
Jul  2 15:16:04.506: Vi1 LCP: TIMEout: State Listen
Jul  2 15:16:04.506: Vi1 LCP: O CONFREQ [Listen] id 168 len 15
Jul  2 15:16:04.506: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:04.506: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:06.506: Vi1 LCP: TIMEout: State REQsent
Jul  2 15:16:06.506: Vi1 LCP: O CONFREQ [REQsent] id 169 len 15
Jul  2 15:16:06.506: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:06.506: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:08.506: Vi1 LCP: TIMEout: State REQsent
Jul  2 15:16:08.506: Vi1 LCP: O CONFREQ [REQsent] id 170 len 15
Jul  2 15:16:08.506: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:08.506: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:10.506: Vi1 LCP: TIMEout: State REQsent
Jul  2 15:16:10.506: Vi1 LCP: O CONFREQ [REQsent] id 171 len 15
Jul  2 15:16:10.506: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:10.506: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:12.507: Vi1 LCP: TIMEout: State REQsent
Jul  2 15:16:12.507: Vi1 LCP: O CONFREQ [REQsent] id 172 len 15
Jul  2 15:16:12.507: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:12.507: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:14.415: Vi1 LCP: I CONFREQ [REQsent] id 88 len 14
Jul  2 15:16:14.415: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul  2 15:16:14.415: Vi1 LCP:    MRU 2048 (0x01040800)
Jul  2 15:16:14.415: Vi1 LCP: O CONFACK [REQsent] id 88 len 14
Jul  2 15:16:14.415: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul  2 15:16:14.415: Vi1 LCP:    MRU 2048 (0x01040800)
Jul  2 15:16:14.507: Vi1 LCP: TIMEout: State ACKsent
Jul  2 15:16:14.507: Vi1 LCP: O CONFREQ [ACKsent] id 173 len 15
Jul  2 15:16:14.507: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:14.507: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:14.539: Vi1 LCP: I CONFACK [ACKsent] id 173 len 15
Jul  2 15:16:14.539: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 15:16:14.539: Vi1 LCP:    MagicNumber 0x121E2132 (0x0506121E2132)
Jul  2 15:16:14.539: Vi1 LCP: State is Open
Jul  2 15:16:14.539: Vi1 PPP: Phase is AUTHENTICATING, by this end [0 sess,
0 load]
Jul  2 15:16:14.539: Vi1 CHAP: O CHALLENGE id 159 len 28 from "nrp1bot"
Jul  2 15:16:14.571: Vi1 CHAP: I RESPONSE id 159 len 34 from "1cardpppoa1_1"
Jul  2 15:16:14.603: Vi1 CHAP: O SUCCESS id 159 len 4
Jul  2 15:16:14.615: Vi1 PPP: Phase is UP [0 sess, 0 load]
Jul  2 15:16:14.615: Vi1 IPCP: O CONFREQ [Not negotiated] id 137 len 10
Jul  2 15:16:14.615: Vi1 IPCP:    Address 10.10.20.13 (0x03060A0A140D)
Jul  2 15:16:14.635: Vi1 IPCP: I CONFREQ [REQsent] id 89 len 10
Jul  2 15:16:14.635: Vi1 IPCP:    Address 0.0.0.0 (0x030600000000)
Jul  2 15:16:14.635: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we
want 0.0.0.0
Jul  2 15:16:14.635: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we
want 0.0.0.0
Jul  2 15:16:14.635: Vi1 IPCP: Pool returned 10.10.20.1
Jul  2 15:16:14.635: Vi1 IPCP: O CONFNAK [REQsent] id 89 len 10
Jul  2 15:16:14.635: Vi1 IPCP:    Address 10.10.20.1 (0x03060A0A1401)
Jul  2 15:16:14.647: Vi1 IPCP: I CONFACK [REQsent] id 137 len 10
Jul  2 15:16:14.647: Vi1 IPCP:    Address 10.10.20.13 (0x03060A0A140D)
Jul  2 15:16:14.667: Vi1 IPCP: I CONFREQ [ACKrcvd] id 90 len 10
Jul  2 15:16:14.667: Vi1 IPCP:    Address 10.10.20.1 (0x03060A0A1401)
Jul  2 15:16:14.667: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 10.10.20.1, we
want 10.10.20.1
Jul  2 15:16:14.667: Vi1 AAA/AUTHOR/IPCP: Reject 10.10.20.1, using
10.10.20.1
Jul  2 15:16:14.667: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 10.10.20.1, we
want 10.10.20.1
Jul  2 15:16:14.667: Vi1 IPCP: O CONFACK [ACKrcvd] id 90 len 10
Jul  2 15:16:14.667: Vi1 IPCP:    Address 10.10.20.1 (0x03060A0A1401)
Jul  2 15:16:14.667: Vi1 IPCP: State is Open
Jul  2 15:16:14.671: Vi1 IPCP: Install route to 10.10.20.1
Jul  2 15:16:15.603: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Virtual-Access1, changed state to up
Jul  2 15:16:24.056: Vi1 LCP: I ECHOREP [Open] id 1 len 12 magic 0xA60C0000
Jul  2 15:16:24.056: Vi1 LCP: Received id 1, sent id 1, line up
nrp1bot#
nrp1bot#
Jul  2 15:16:29.412: Vi1 LCP: I ECHOREQ [Open] id 91 len 8 magic 0xA60C0000
Jul  2 15:16:29.412: Vi1 LCP: O ECHOREP [Open] id 91 len 8 magic 0x121E2132
nrp1bot#
```

Remote User Initiates a PPPoE Session (Step 1 and Step 2)

The remote user initiates a Point-to-Point Protocol over Ethernet (PPPoE) session over the DSL access network. If the remote user cannot initiate a PPPoE session, use the following **debug** commands in privileged EXEC mode:

- **debug ppp negotiation**—Displays PPP packets transmitted during PPP startup, where PPP options are negotiated.
- **debug ppp authentication**—Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.
- **debug vpdn pppoe-events**—Displays PPPoE protocol errors that prevent a session from being established or errors that cause an established session to be closed.

Example 3-18 provides a sample of the **debug** command output that results from these commands.

Example 3-18 Sample Debug DSL Router Cannot Initiate PPPoE Session

```
PPPoE protocol events debugging is on

nrp1bot# debug ppp authentication

PPP authentication debugging is on

nrp1bot# debug ppp negotiation

PPP protocol negotiation debugging is on

nrp1bot# sh debug


PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
VPN:
  PPPoE protocol events debugging is on


nrp1bot#
Jul  2 19:07:56.686: PPPoE: IN PADI discovery packet
Jul  2 19:07:56.686:   PPPoE: LMAC:ffff.ffff.ffff RMAC:00a0.ccd9.2697 21/33 
AT0/0/0.2133
Jul  2 19:07:56.686: PPPoE: PADO OUT from PPPoE tunnel
Jul  2 19:07:56.686:   PPPoE: LMAC:0002.b992.7807 RMAC:00a0.ccd9.2697 21/33 
AT0/0/0.2133
Jul  2 19:07:56.738: PPPoE: IN PADR discovery packet
Jul  2 19:07:56.738:   PPPoE: LMAC:0002.b992.7807 RMAC:00a0.ccd9.2697 21/33 
AT0/0/0.2133
Jul  2 19:07:56.738: Vi2 PPP: Phase is DOWN, Setup [0 sess, 0 load]
Jul  2 19:07:56.758: PPPoE: Create session: 1
Jul  2 19:07:56.758: PPPoE:  1: Created
Jul  2 19:07:56.758:   PPPoE: LMAC:0002.b992.7807 RMAC:00a0.ccd9.2697 21/33 
AT0/0/0.2133
Jul  2 19:07:56.758: PPPoE: PADS OUT from PPPoE tunnel
Jul  2 19:07:56.758:   PPPoE: LMAC:0002.b992.7807 RMAC:00a0.ccd9.2697 21/33 
AT0/0/0.2133
Jul  2 19:07:56.762: %LINK-3-UPDOWN: Interface Virtual-Access2, changed 
state to up
Jul  2 19:07:56.762: Vi2 PPP: Treating connection as a dedicated line
Jul  2 19:07:56.762: Vi2 PPP: Phase is ESTABLISHING, Active Open [0 sess, 0 
load]
Jul  2 19:07:56.762: Vi2 LCP: O CONFREQ [Closed] id 1 len 19
Jul  2 19:07:56.762: Vi2 LCP:    MRU 1492 (0x010405D4)
Jul  2 19:07:56.762: Vi2 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 19:07:56.762: Vi2 LCP:    MagicNumber 0x12F26578 (0x050612F26578)
Jul  2 19:07:56.918: Vi2 LCP: I CONFREQ [REQsent] id 1 len 14
Jul  2 19:07:56.918: Vi2 LCP:    MagicNumber 0x001B0B5A (0x0506001B0B5A)
Jul  2 19:07:56.918: Vi2 LCP:    PFC (0x0702)
Jul  2 19:07:56.918: Vi2 LCP:    ACFC (0x0802)
Jul  2 19:07:56.918: Vi2 LCP: O CONFACK [REQsent] id 1 len 14
Jul  2 19:07:56.918: Vi2 LCP:    MagicNumber 0x001B0B5A (0x0506001B0B5A)
Jul  2 19:07:56.918: Vi2 LCP:    PFC (0x0702)
Jul  2 19:07:56.918: Vi2 LCP:    ACFC (0x0802)
Jul  2 19:07:58.762: Vi2 LCP: TIMEout: State ACKsent
Jul  2 19:07:58.762: Vi2 LCP: O CONFREQ [ACKsent] id 2 len 19
Jul  2 19:07:58.762: Vi2 LCP:    MRU 1492 (0x010405D4)
Jul  2 19:07:58.762: Vi2 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 19:07:58.762: Vi2 LCP:    MagicNumber 0x12F26578 (0x050612F26578)
Jul  2 19:07:58.798: Vi2 LCP: I CONFACK [ACKsent] id 2 len 19
Jul  2 19:07:58.798: Vi2 LCP:    MRU 1492 (0x010405D4)
Jul  2 19:07:58.798: Vi2 LCP:    AuthProto CHAP (0x0305C22305)
Jul  2 19:07:58.798: Vi2 LCP:    MagicNumber 0x12F26578 (0x050612F26578)
Jul  2 19:07:58.798: Vi2 LCP: State is Open
Jul  2 19:07:58.798: Vi2 PPP: Phase is AUTHENTICATING, by this end [0 sess,
0 load]
Jul  2 19:07:58.798: Vi2 CHAP: O CHALLENGE id 11 len 28 from "nrp1bot"
Jul  2 19:07:58.838: Vi2 CHAP: I RESPONSE id 11 len 34 from "1cardpppoe1_1"
Jul  2 19:07:58.838: Vi2 PPP: Phase is FORWARDING [0 sess, 0 load]
Jul  2 19:07:58.838: Vi2 PPP: Phase is AUTHENTICATING [0 sess, 0 load]
Jul  2 19:07:58.850: Vi2 CHAP: O SUCCESS id 11 len 4
Jul  2 19:07:58.870: Vi2 PPP: Phase is UP [0 sess, 0 load]
Jul  2 19:07:58.870: Vi2 IPCP: O CONFREQ [Closed] id 1 len 10
Jul  2 19:07:58.870: Vi2 IPCP:    Address 10.10.21.37 (0x03060A0A1525)
Jul  2 19:07:58.886: Vi2 IPCP: I CONFREQ [REQsent] id 1 len 34
Jul  2 19:07:58.886: Vi2 IPCP:    Address 0.0.0.0 (0x030600000000)
Jul  2 19:07:58.886: Vi2 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Jul  2 19:07:58.886: Vi2 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Jul  2 19:07:58.886: Vi2 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Jul  2 19:07:58.890: Vi2 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Jul  2 19:07:58.890: Vi2 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we
want 0.0.0.0
Jul  2 19:07:58.890: Vi2 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we
want 0.0.0.0
Jul  2 19:07:58.890: Vi2 IPCP: Pool returned 10.10.21.26
Jul  2 19:07:58.890: Vi2 IPCP: O CONFREJ [REQsent] id 1 len 28
Jul  2 19:07:58.890: Vi2 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Jul  2 19:07:58.890: Vi2 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Jul  2 19:07:58.890: Vi2 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Jul  2 19:07:58.890: Vi2 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Jul  2 19:07:58.906: Vi2 IPCP: I CONFACK [REQsent] id 1 len 10
Jul  2 19:07:58.906: Vi2 IPCP:    Address 10.10.21.37 (0x03060A0A1525)
Jul  2 19:07:58.926: Vi2 IPCP: I CONFREQ [ACKrcvd] id 2 len 10
Jul  2 19:07:58.926: Vi2 IPCP:    Address 0.0.0.0 (0x030600000000)
Jul  2 19:07:58.926: Vi2 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we
want 10.10.21.26
Jul  2 19:07:58.926: Vi2 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we
want 10.10.21.26
Jul  2 19:07:58.926: Vi2 IPCP: O CONFNAK [ACKrcvd] id 2 len 10
Jul  2 19:07:58.926: Vi2 IPCP:    Address 10.10.21.26 (0x03060A0A151A)
Jul  2 19:07:58.962: Vi2 IPCP: I CONFREQ [ACKrcvd] id 3 len 10
Jul  2 19:07:58.962: Vi2 IPCP:    Address 10.10.21.26 (0x03060A0A151A)
Jul  2 19:07:58.962: Vi2 AAA/AUTHOR/IPCP: Start.  Her address 10.10.21.26,
we want 10.10.21.26
Jul  2 19:07:58.966: Vi2 AAA/AUTHOR/IPCP: Reject 10.10.21.26, using
10.10.21.26
Jul  2 19:07:58.966: Vi2 AAA/AUTHOR/IPCP: Done.  Her address 10.10.21.26, we
want 10.10.21.26
Jul  2 19:07:58.966: Vi2 IPCP: O CONFACK [ACKrcvd] id 3 len 10
Jul  2 19:07:58.966: Vi2 IPCP:    Address 10.10.21.26 (0x03060A0A151A)
Jul  2 19:07:58.966: Vi2 IPCP: State is Open
Jul  2 19:07:58.966: Vi2 IPCP: Install route to 10.10.21.26
Jul  2 19:07:59.850: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Virtual-Access2, changed state to up
```

VHG/PE Queries SP RADIUS Server (Step 3 and Step 4[a])

The VHG-PE queries RADIUS to associate the remote user with a specific customer MPLS VPN.

Note The VPN's VRF (routing table and other information associated with a specific VPN) must have been pre-instantiated on the VHG/PE.

If the VHG/PE cannot query the SP RADIUS server, use the following **debug** commands in privileged EXEC mode:

- **debug aaa authentication**—Displays information on AAA/Terminal Access Controller Access Control System Plus (TACACS+) authentication.
- **debug aaa authorization**—Displays information on AAA/TACACS+ authorization. To disable debugging output, use the **no** form of this command.
- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).

Example 3-19 provides a sample of the **debug** command output that results from these commands.

Example 3-19 Sample Debug VHG/PE Queries SP RADIUS Server

```
nrp1bot# debug aaa authentication

AAA Authentication debugging is on

nrp1bot# debug aaa authorization

AAA Authorization debugging is on

nrp1bot# debug radius

Radius protocol debugging is on
nrp1bot#

nrp1bot# sh debug

General OS:
  AAA Authentication debugging is on
  AAA Authorization debugging is on

Radius protocol debugging is on
nrp1bot#
Jul  2 15:47:13.462: AAA: parse name=Virtual-Access1 idb type=21 tty=-1
Jul  2 15:47:13.462: AAA: name=Virtual-Access1 flags=0x11 type=6 shelf=0
slot=0 adapter=0 port=1 channel=0
Jul  2 15:47:13.462: AAA/MEMORY: create_user (0x6167DD48)
user='1cardpppoa1_1' ruser='' port='Virtual-Access1' rem_addr=''
authen_type=CHAP service=PPP priv=1
Jul  2 15:47:13.462: AAA/AUTHEN/START (1344226464): port='Virtual-Access1'
list='' action=LOGIN service=PPP
Jul  2 15:47:13.462: AAA/AUTHEN/START (1344226464): using "default" list
Jul  2 15:47:13.462: AAA/AUTHEN/START (1344226464): Method=radius (radius)
Jul  2 15:47:13.462: RADIUS: ustruct sharecount=1
Jul  2 15:47:13.462: RADIUS: Initial Transmit Virtual-Access1 id 159
172.29.51.235:1645, Access-Request, len 84
Jul  2 15:47:13.462:         Attribute 4 6 81010109
Jul  2 15:47:13.462:         Attribute 5 6 00000000
Jul  2 15:47:13.462:         Attribute 61 6 00000005
Jul  2 15:47:13.462:         Attribute 1 15 31636172
Jul  2 15:47:13.462:         Attribute 3 19 A0DFA8B1
Jul  2 15:47:13.462:         Attribute 6 6 00000002
Jul  2 15:47:13.462:         Attribute 7 6 00000001
Jul  2 15:47:13.474: RADIUS: Received from id 159 172.29.51.235:1645,
Access-Accept, len 237
Jul  2 15:47:13.474:         Attribute 6 6 00000002
Jul  2 15:47:13.474:         Attribute 7 6 00000001
Jul  2 15:47:13.474:         Attribute 26 12 00000211F4060000
Jul  2 15:47:13.474:         Attribute 26 62 0000000901386C63
Jul  2 15:47:13.474:         Attribute 26 51 00000009012D6C63
Jul  2 15:47:13.474:         Attribute 26 80 00000009014A6C63
Jul  2 15:47:13.474: AAA/AUTHEN (1344226464): status = PASS
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP: Authorize LCP
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP (2669002508): Port='Virtual-Access1'
list='' service=NET
Jul  2 15:47:13.474: AAA/AUTHOR/LCP: Vi1 (2669002508) user='1cardpppoa1_1'
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP (2669002508): send AV service=ppp
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP (2669002508): send AV protocol=lcp
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP (2669002508): found list "default"
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP (2669002508): Method=radius (radius)
Jul  2 15:47:13.474: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf
forwarding 1cardpppoa1"
Jul  2 15:47:13.474: RADIUS: cisco AVPair "lcp:interface-config#2 = ip
unnumbered lo20"
Jul  2 15:47:13.474: RADIUS: cisco AVPair "lcp:interface-config#3 = peer
default ip address pool 1cardpppoa1_1_pool"
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR (2669002508): Post authorization status
= PASS_REPL
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP: Processing AV service=ppp
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP: Processing AV idletime=1800
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP: Processing AV interface-config#1 =
ip vrf forwarding 1cardpppoa1
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP: Processing AV interface-config#2 =
ip unnumbered lo20
Jul  2 15:47:13.474: Vi1 AAA/AUTHOR/LCP: Processing AV interface-config#3 =
peer default ip address pool 1cardpppoa1_1_pool
Jul  2 15:47:13.478: AAA/AUTHOR/LCP: Virtual-Access1: Reconstruct
interface-config= ip vrf forwarding 1cardpppoa1\n ip unnumbered lo20\n peer
default ip address pool 1cardpppoa1_1_pool

Jul  2 15:47:13.478: Vi1 AAA/AUTHOR/LCP: Per-user interface config created:
ppp timeout idle 1800 aaa
 ip vrf forwarding 1cardpppoa1\n ip unnumbered lo20\n peer default ip
address pool 1cardpppoa1_1_pool

Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM: (0): Can we start IPCP?
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM (907722013): Port='Virtual-Access1'
list='' service=NET
Jul  2 15:47:13.490: AAA/AUTHOR/FSM: Vi1 (907722013) user='1cardpppoa1_1'
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM (907722013): send AV service=ppp
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM (907722013): send AV protocol=ip
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM (907722013): found list "default"
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM (907722013): Method=radius (radius)
Jul  2 15:47:13.490: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf
forwarding 1cardpppoa1" not applied for ip
Jul  2 15:47:13.490: RADIUS: cisco AVPair "lcp:interface-config#2 = ip
unnumbered lo20" not applied for ip
Jul  2 15:47:13.490: RADIUS: cisco AVPair "lcp:interface-config#3 = peer
default ip address pool 1cardpppoa1_1_pool" not applied for ip
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR (907722013): Post authorization status =
PASS_REPL
Jul  2 15:47:13.490: Vi1 AAA/AUTHOR/FSM: We can start IPCP
Jul  2 15:47:13.510: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we
want 0.0.0.0
Jul  2 15:47:13.510: Vi1 AAA/AUTHOR/IPCP: Processing AV service=ppp
Jul  2 15:47:13.510: Vi1 AAA/AUTHOR/IPCP: Authorization succeeded
Jul  2 15:47:13.510: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we
want 0.0.0.0
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 10.10.20.1, we
want 10.10.20.1
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP (1098858913):
Port='Virtual-Access1' list='' service=NET
Jul  2 15:47:13.542: AAA/AUTHOR/IPCP: Vi1 (1098858913) user='1cardpppoa1_1'
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP (1098858913): send AV service=ppp
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP (1098858913): send AV protocol=ip
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP (1098858913): send AV
addr*10.10.20.1
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP (1098858913): found list "default"
Jul  2 15:47:13.542: Vi1 AAA/AUTHOR/IPCP (1098858913): Method=radius
(radius)
Jul  2 15:47:13.542: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf
forwarding 1cardpppoa1" not applied for ip
Jul  2 15:47:13.546: RADIUS: cisco AVPair "lcp:interface-config#2 = ip
unnumbered lo20" not applied for ip
Jul  2 15:47:13.546: RADIUS: cisco AVPair "lcp:interface-config#3 = peer
default ip address pool 1cardpppoa1_1_pool" not applied for ip
Jul  2 15:47:13.546: Vi1 AAA/AUTHOR (1098858913): Post authorization status
= PASS_REPL
Jul  2 15:47:13.546: Vi1 AAA/AUTHOR/IPCP: Reject 10.10.20.1, using
10.10.20.1
Jul  2 15:47:13.546: Vi1 AAA/AUTHOR/IPCP: Processing AV service=ppp
Jul  2 15:47:13.546: Vi1 AAA/AUTHOR/IPCP: Processing AV addr*10.10.20.1
Jul  2 15:47:13.546: Vi1 AAA/AUTHOR/IPCP: Authorization succeeded
Jul  2 15:47:13.546: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 10.10.20.1, we
want 10.10.20.1
Jul  2 15:47:14.478: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Virtual-Access1, changed state to up
```

Remote User Is Associated with a Specific Customer MPLS VPN (Step 3 and Step 4[b])

If the remote user cannot be associated with a specific customer MPLS VPN, use the following **debug** commands in privileged EXEC mode:

- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).
- **debug vtemplate**—Displays cloning information for a virtual access interface from the time it is cloned from a virtual template to the time the virtual access interface comes down when the call ends.

Example 3-20 provides a sample of the **debug** command output that results from these commands.

Example 3-20 Sample Debug Remote User Is Associated with a Specific Customer MPLS VPN

```
nrp1bot# debug radius

Radius protocol debugging is on

nrp1bot# debug vtemplate

Virtual Template debugging is on

nrp1bot#
Jul  2 15:51:47.426: RADIUS: ustruct sharecount=1
Jul  2 15:51:47.426: RADIUS: Initial Transmit Virtual-Access1 id 160
172.29.51.235:1645, Access-Request, len 84
Jul  2 15:51:47.426:         Attribute 4 6 81010109
Jul  2 15:51:47.426:         Attribute 5 6 00000000
Jul  2 15:51:47.426:         Attribute 61 6 00000005
Jul  2 15:51:47.426:         Attribute 1 15 31636172
Jul  2 15:51:47.426:         Attribute 3 19 A18162B8
Jul  2 15:51:47.426:         Attribute 6 6 00000002
Jul  2 15:51:47.426:         Attribute 7 6 00000001
Jul  2 15:51:47.434: RADIUS: Received from id 160 172.29.51.235:1645,
Access-Accept, len 237
```

```
Jul  2 15:51:47.434:         Attribute 6 6 00000002
Jul  2 15:51:47.434:         Attribute 7 6 00000001
Jul  2 15:51:47.434:         Attribute 26 12 00000211F4060000
Jul  2 15:51:47.434:         Attribute 26 62 0000000901386C63
Jul  2 15:51:47.438:         Attribute 26 51 00000009012D6C63
Jul  2 15:51:47.438:         Attribute 26 80 00000009014A6C63
Jul  2 15:51:47.438: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf
forwarding 1cardpppoa1"
Jul  2 15:51:47.438: RADIUS: cisco AVPair "lcp:interface-config#2 = ip
unnumbered lo20"
Jul  2 15:51:47.438: RADIUS: cisco AVPair "lcp:interface-config#3 = peer
default ip address pool 1cardpppoa1_1_pool"
Jul  2 15:51:47.438: Vi1 VTEMPLATE: Has a new cloneblk AAA, now it has
vtemplate/AAA
Jul  2 15:51:47.438: Vi1 VTEMPLATE:
Jul  2 15:51:47.438: Vi1 VTEMPLATE: Clone from AAA
interface Virtual-Access1
ppp timeout idle 1800 aaa
 ip vrf forwarding 1cardpppoa1
 ip unnumbered lo20
 peer default ip address pool 1cardpppoa1_1_pool
end

Jul  2 15:51:47.450: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf
forwarding 1cardpppoa1" not applied for ip
Jul  2 15:51:47.450: RADIUS: cisco AVPair "lcp:interface-config#2 = ip
unnumbered lo20" not applied for ip
Jul  2 15:51:47.450: RADIUS: cisco AVPair "lcp:interface-config#3 = peer
default ip address pool 1cardpppoa1_1_pool" not applied for ip
Jul  2 15:51:47.506: RADIUS: cisco AVPair "lcp:interface-config#1 = ip vrf
forwarding 1cardpppoa1" not applied for ip
Jul  2 15:51:47.506: RADIUS: cisco AVPair "lcp:interface-config#2 = ip
unnumbered lo20" not applied for ip
Jul  2 15:51:47.506: RADIUS: cisco AVPair "lcp:interface-config#3 = peer
default ip address pool 1cardpppoa1_1_pool" not applied for ip
Jul  2 15:51:48.438: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Virtual-Access1, changed state to up
nrp1bot#
```

VHG/PE Obtains IP Address (Step 5)

If the VHG/PE cannot obtain an IP address, use the following **debug** command in privileged EXEC mode:

- **debug ip peer**—Displays address activity and contains additional output when pool groups are defined.

Example 3-21 provides a sample of the **debug** command output that results from this command.

Example 3-21 Sample Debug VHG/PE Obtains IP Address

```
nrp1bot# debug ip peer

IP peer address activity debugging is on

nrp1bot# sh debug

Generic IP:
  IP peer address activity debugging is on

nrp1bot#
Jul  2 16:02:20.497: Vi1: Pools to search : 1cardpppoa1_1_pool
Jul  2 16:02:20.497: Vi1: Pool 1cardpppoa1_1_pool returned address =
10.10.20.1
Jul  2 16:02:20.529: Vi1 IPCP: Install route to 10.10.20.1
Jul  2 16:02:21.461: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Virtual-Access1, changed state to up
```

Verifying Correct Configuration for PPPoX Remote Access to MPLS VPN Integration

The following **show** commands are useful in debugging PPPoX to MPLS VPN integration:

- **show atm pvc ppp**—Displays all ATM permanent virtual circuits (PVCs) and traffic information; applies only to PPPoA.
- **show int virtual-access** *virtual access interface #*—Displays status, traffic data, and configuration information about a specified virtual access interface.
- **show ip route vrf** *vrf name*—Display the IP routing table associated with a VRF (VPN routing/forwarding instance).
- **show ip local pool**—Displays statistics for any defined IP address pools.
- **show vpdn session** [**all**]—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network (VPDN); PPPoE specific.
- **show vpdn tunnel**—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network (VPDN); PPPoE specific.

Example 3-22 shows the detailed output that results when you implement these **show** commands.

Example 3-22 Sample show Command Output for PPPoX Remote Access to MPLS VPN Integration

```
nrp1bot# sh atm pvc ppp

           VCD /                                        Peak Avg/Min Burst
ATM Int.   Name        VPI   VCI  Type      VA VASt  SC    Kbps  Kbps   Cells VCSt
0/0/0.2033 5            20    33   PVC       1   UP UBR  155000                UP

nrp1bot# sh int virtual-access 2

Virtual-Access2 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback21 (10.10.21.37)
  MTU 1492 bytes, BW 100000 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  Time to interface disconnect:  idle 00:29:28
  LCP Open
  Open: IPCP
  Bound to ATM0/0/0.2133 VCD: 6, VPI: 21, VCI: 33
  Cloned from virtual-template: 2
  Last input 00:00:07, output never, output hang never
  Last clearing of "show interface" counters 06:11:22
  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
     157 packets input, 9097 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
     12 packets output, 226 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions

nrp1bot# sh ip route vrf 1cardpppoe1

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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/32 is subnetted, 3 subnets
C       10.10.21.26 is directly connected, Virtual-Access2
C       10.10.21.37 is directly connected, Loopback21
B       10.10.21.38 [200/0] via 10.1.1.4, 07:17:30

nrp1bot# sh ip local pool

 Pool                     Begin           End             Free  In use
 1cardpppoa1_1_pool       10.10.20.1      10.10.20.6         6       0
 1cardpppoe1_1_pool       10.10.21.26     10.10.21.30        4       1

nrp1bot# sh vpdn session all

%No active L2TP tunnels

%No active L2F tunnels

%No active PPTP tunnels

PPPoE Session Information Total tunnels 1 sessions 1

session id: 1
local MAC address: 0002.b992.7807, remote MAC address: 00a0.ccd9.2697
virtual access interface: Vi2, outgoing interface: AT0/0/0, vc: 21/33
    16 packets sent, 161 received
    258 bytes sent, 8887 received

nrp1bot#sh vpdn tunnel

%No active L2TP tunnels

%No active L2F tunnels

%No active PPTP tunnels

PPPoE Tunnel Information Total tunnels 1 sessions 1

PPPoE Tunnel Information

Session count: 1
nrp1bot#
```

PPPoX Remote Access SSG to MPLS VPN Integration

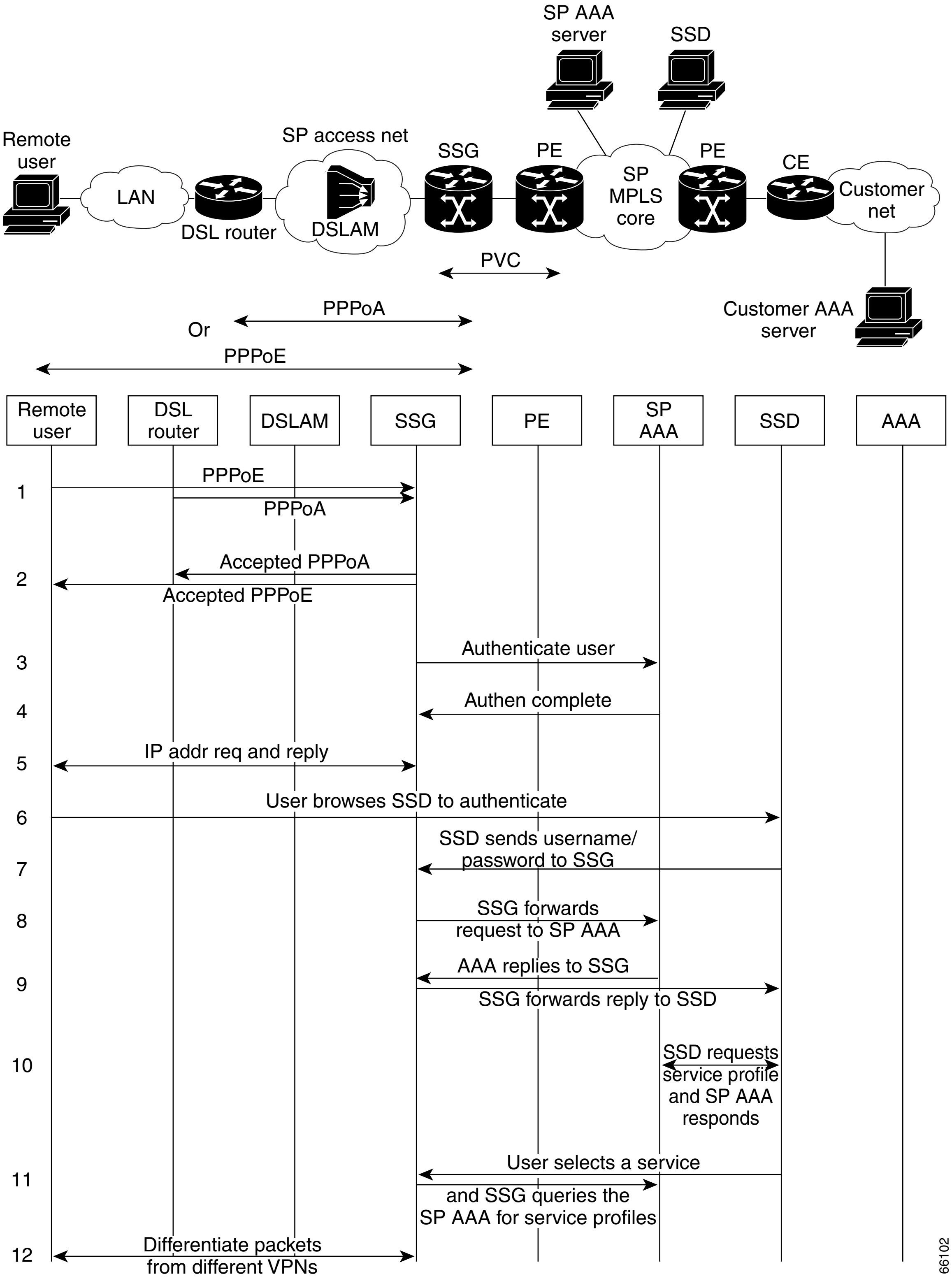
This section contains the following troubleshooting topics:

- Overview of DSL PPPOX Remote Access SSG to MPLS VPN Integration, page 3-26
- Initiating and Viewing debug Command Output, page 3-28
- Debugging Problems Associated with PPPoX Remote Access SSG to MPLS VPN Integration, page 3-28, including:
  - DSL Router Cannot Initiate PPPoX Session (Step 1 and Step 2), page 3-28
  - SSG Queries the RADIUS Server (Step 3 and Step 4[a]), page 3-29
  - SSG Completes the Remote User's Authentication (Step 3 and Step 4[b]), page 3-32
  - SSG Obtains an IP Address (Step 5), page 3-33
  - SSG User Browses SSD and Authenticates (Step 6 through Step 10), page 3-34
  - SSG User Selects a Service From SSD (Step 11), page 3-36
- Verifying Correct Configuration for PPPoX Remote Access SSG to MPLS VPN Integration, page 3-38

Overview of DSL PPPOX Remote Access SSG to MPLS VPN Integration

Figure 3-4 shows the topology associated with a VPN-capable service provider's MPLS backbone. In this scenario, you should assume that the customer has outsourced all remote access operations to its service provider.

Figure 3-4 DSL PPPoX Remote Access



Initiating and Viewing debug Command Output

For reminders on using the command- line interface for viewing debug output, refer to the sections of the *Remote Access to MPLS VPN Solution Provisioning Guide 1.0* entitled "User Interface Command Modes" and "User Command Modes." This document can be accessed at the following URL:

http://cco/univercd/cc/td/doc/product/vpn/solution/rampls/index.htm

Debugging Problems Associated with PPPoX Remote Access SSG to MPLS VPN Integration

Table 3-4 (below) corresponds to Figure 3-4, providing a cross-reference to the troubleshooting topics associated with the events that occur when the remote user creates a PPPoX session over DSL in an attempt to access its corporate network or ISP.

Table 3-4 Troubleshooting Topics for DSL PPPOX to MPLS VPN Integration

Event Shown in Figure 3-4:	Related Troubleshooting Topic(s):
Line1-2: DSL router initiates a PPPoA or PPPoE session. SSG accepts and terminates the PPPoX session.	DSL Router Cannot Initiate PPPoX Session (Step 1 and Step 2), page 3-28
Lines 3-4: SSG queries the RADIUS server for attributes of incoming user. SSG completes the remote user's authentication via RADIUS.	SSG Queries the RADIUS Server (Step 3 and Step 4[a]), page 3-29 SSG Completes the Remote User's Authentication (Step 3 and Step 4[b]), page 3-32
Line 5: SSG obtains an IP address for the remote user.	SSG Obtains an IP Address (Step 5), page 3-33
Lines 6-10: SSG user browses SSD and authenticates.	SSG User Browses SSD and Authenticates (Step 6 through Step 10), page 3-34
Line 11: SSG user selects a service from SSD.	SSG User Selects a Service From SSD (Step 11), page 3-36

DSL Router Cannot Initiate PPPoX Session (Step 1 and Step 2)

If the DSL router cannot initiate a PPPoX session, use the following debug commands in privileged EXEC mode:

- **debug ppp negotiation**—Displays PPP packets transmitted during PPP startup, where PPP options are negotiated.
- **debug ppp authentication**—Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.
- **debug vpdn pppoe-events**—Displays PPPoE protocol errors that prevent a session from being established or errors that cause an established session to be closed.

Example 3-23 provides a sample of the **debug** command output that results from these commands.

Example 3-23 Sample Debug DSL Router Cannot Initiate PPPoX Session

```
nrp1mid# debug ppp negotiation

PPP protocol negotiation debugging is on

nrp1mid# debug ppp authentication

PPP authentication debugging is on

nrp1mid# sh debug

PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on

Jul 25 15:02:02.903: Vi1 LCP: I CONFREQ [Listen] id 253 len 14
Jul 25 15:02:02.903: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul 25 15:02:02.903: Vi1 LCP:    MRU 2048 (0x01040800)
Jul 25 15:02:02.903: Vi1 LCP: O CONFREQ [Listen] id 58 len 15
Jul 25 15:02:02.903: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul 25 15:02:02.903: Vi1 LCP:    MagicNumber 0x06AA8DAF (0x050606AA8DAF)
Jul 25 15:02:02.903: Vi1 LCP: O CONFACK [Listen] id 253 len 14
Jul 25 15:02:02.903: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul 25 15:02:02.903: Vi1 LCP:    MRU 2048 (0x01040800)
Jul 25 15:02:02.935: Vi1 LCP: I CONFACK [ACKsent] id 58 len 15
Jul 25 15:02:02.935: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul 25 15:02:02.935: Vi1 LCP:    MagicNumber 0x06AA8DAF (0x050606AA8DAF)
Jul 25 15:02:02.935: Vi1 LCP: State is Open
Jul 25 15:02:02.935: Vi1 PPP: Phase is AUTHENTICATING, by this end [0 sess, 0 load]
Jul 25 15:02:02.935: Vi1 CHAP: O CHALLENGE id 7 len 28 from "nrp1mid"
Jul 25 15:02:02.967: Vi1 CHAP: I RESPONSE id 7 len 32 from "2cardpppoa1"
Jul 25 15:02:02.967: Vi1 PPP: Phase is FORWARDING [0 sess, 0 load]
Jul 25 15:02:02.967: Vi1 PPP: Phase is AUTHENTICATING [0 sess, 0 load]
Jul 25 15:02:02.979: Vi1 CHAP: O SUCCESS id 7 len 4
Jul 25 15:02:02.979: Vi1 PPP: Phase is UP [0 sess, 0 load]
Jul 25 15:02:02.979: Vi1 IPCP: O CONFREQ [Closed] id 5 len 10
Jul 25 15:02:02.979: Vi1 IPCP:    Address 10.1.1.6 (0x030681010106)
Jul 25 15:02:03.015: Vi1 IPCP: I CONFREQ [REQsent] id 254 len 10
Jul 25 15:02:03.015: Vi1 IPCP:    Address 0.0.0.0 (0x030600000000)
Jul 25 15:02:03.015: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 0.0.0.0
Jul 25 15:02:03.015: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we want 10.1.4.1
Jul 25 15:02:03.015: Vi1 IPCP: O CONFNAK [REQsent] id 254 len 10
Jul 25 15:02:03.015: Vi1 IPCP:    Address 10.1.4.1 (0x030681010401)
Jul 25 15:02:03.019: Vi1 IPCP: I CONFACK [REQsent] id 5 len 10
Jul 25 15:02:03.019: Vi1 IPCP:    Address 10.1.1.6 (0x030681010106)
Jul 25 15:02:03.051: Vi1 IPCP: I CONFREQ [ACKrcvd] id 255 len 10
Jul 25 15:02:03.051: Vi1 IPCP:    Address 10.1.4.1 (0x030681010401)
Jul 25 15:02:03.051: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 10.1.4.1, we want 10.1.4.1
Jul 25 15:02:03.051: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 10.1.4.1, we want 10.1.4.1
Jul 25 15:02:03.051: Vi1 IPCP: O CONFACK [ACKrcvd] id 255 len 10
Jul 25 15:02:03.051: Vi1 IPCP:    Address 10.1.4.1 (0x030681010401)
Jul 25 15:02:03.051: Vi1 IPCP: State is Open
Jul 25 15:02:03.055: Vi1 IPCP: Install route to 10.1.4.1
Jul 25 15:02:03.979: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp1mid#
```

SSG Queries the RADIUS Server (Step 3 and Step 4[a])

Note The VPN's VRF (routing table and other information associated with a specific VPN) must have been pre-instantiated on the VHG/PE.

If SSG cannot query the RADIUS server, use the following **debug** commands in privileged EXEC mode:

- **debug aaa authentication**—Displays information on AAA/Terminal Access Controller Access Control System Plus (TACACS+) authentication.
- **debug aaa authorization**—Displays information on AAA/TACACS+ authorization. To disable debugging output, use the **no** form of this command.
- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).

Example 3-24 provides a sample of the **debug** command output that results from these commands.

Example 3-24 Sample Debug SSG Queries the RADIUS Server

```
nrp1mid# debug aaa authentication

AAA Authentication debugging is on

nrp1mid# debug radius

Radius protocol debugging is on

nrp1mid# debug aaa authorization

AAA Authorization debugging is on

nrp1mid# sh debug

General OS:
  AAA Authentication debugging is on
  AAA Authorization debugging is on
Radius protocol debugging is on
nrp1mid#
Jul 25 15:37:35.963: AAA: parse name=Virtual-Access1 idb type=21 tty=-1
Jul 25 15:37:35.963: AAA: name=Virtual-Access1 flags=0x11 type=6 shelf=0 slot=0 adapter=0
port=1 channel=0
Jul 25 15:37:35.963: AAA/MEMORY: create_user (0x616AD95C) user='2cardpppoa1' ruser=''
port='Virtual-Access1' rem_addr='' authen_type=CHAP service=PPP priv=1
Jul 25 15:37:35.963: RADIUS: ustruct sharecount=1
Jul 25 15:37:35.963: RADIUS: Initial Transmit Virtual-Access1 id 180 172.29.51.235:1645,
Access-Request, len 82
Jul 25 15:37:35.963:         Attribute 4 6 81010106
Jul 25 15:37:35.967:         Attribute 5 6 00000001
Jul 25 15:37:35.967:         Attribute 61 6 00000005
Jul 25 15:37:35.967:         Attribute 1 13 32636172
Jul 25 15:37:35.967:         Attribute 3 19 0B6D884F
Jul 25 15:37:35.967:         Attribute 6 6 00000002
Jul 25 15:37:35.967:         Attribute 7 6 00000001
Jul 25 15:37:35.975: RADIUS: Received from id 180 172.29.51.235:1645, Access-Accept, len
50
Jul 25 15:37:35.975:         Attribute 6 6 00000002
Jul 25 15:37:35.975:         Attribute 7 6 00000001
Jul 25 15:37:35.975:         Attribute 8 6 81010401
Jul 25 15:37:35.975:         Attribute 9 6 FFFFFFF8
Jul 25 15:37:35.975:         Attribute 12 6 000005DC
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP: Authorize LCP
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP (1705808688): Port='Virtual-Access1' list=''
service=NET
Jul 25 15:37:35.975: AAA/AUTHOR/LCP: Vi1 (1705808688) user='2cardpppoa1'
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP (1705808688): send AV service=ppp
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP (1705808688): send AV protocol=lcp
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP (1705808688): found list "default"
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP (1705808688): Method=radius (radius)
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR (1705808688): Post authorization status = PASS_REPL
Jul 25 15:37:35.975: Vi1 AAA/AUTHOR/LCP: Processing AV service=ppp
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM: (0): Can we start IPCP?
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM (3853248361): Port='Virtual-Access1' list=''
service=NET
Jul 25 15:37:35.979: AAA/AUTHOR/FSM: Vi1 (3853248361) user='2cardpppoa1'
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM (3853248361): send AV service=ppp
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM (3853248361): send AV protocol=ip
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM (3853248361): found list "default"
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM (3853248361): Method=radius (radius)
Jul 25 15:37:35.979: RADIUS: Authorize IP address 10.10.20.1
Jul 25 15:37:35.979: RADIUS: Framed-IP-Netmask 255.255.255.248
Jul 25 15:37:35.979: RADIUS: framed-route 10.1.4.0 255.255.255.248
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR (3853248361): Post authorization status = PASS_REPL
Jul 25 15:37:35.979: Vi1 AAA/AUTHOR/FSM: We can start IPCP
Jul 25 15:37:35.979: RADIUS: ustruct sharecount=3
Jul 25 15:37:35.979: RADIUS: Initial Transmit Virtual-Access1 id 181 172.29.51.235:1646,
Accounting-Request, len 85
Jul 25 15:37:35.979:         Attribute 4 6 81010106
Jul 25 15:37:35.979:         Attribute 5 6 00000001
Jul 25 15:37:35.979:         Attribute 61 6 00000005
Jul 25 15:37:35.979:         Attribute 1 13 32636172
Jul 25 15:37:35.979:         Attribute 40 6 00000001
Jul 25 15:37:35.979:         Attribute 6 6 00000002
Jul 25 15:37:35.979:         Attribute 44 10 30303030
Jul 25 15:37:35.979:         Attribute 7 6 00000001
Jul 25 15:37:35.979:         Attribute 41 6 00000000
Jul 25 15:37:36.015: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 0.0.0.0
Jul 25 15:37:36.015: Vi1 AAA/AUTHOR/IPCP: Processing AV service=ppp
Jul 25 15:37:36.015: Vi1 AAA/AUTHOR/IPCP: Processing AV addr=10.1.4.1
Jul 25 15:37:36.015: Vi1 AAA/AUTHOR/IPCP: Processing AV netmask*255.255.255.248
Jul 25 15:37:36.015: Vi1 AAA/AUTHOR/IPCP: Processing AV route=10.1.4.0 255.255.255.248
Jul 25 15:37:36.015: Vi1 AAA/AUTHOR/IPCP: Authorization succeeded
Jul 25 15:37:36.019: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we want 10.1.4.1
Jul 25 15:37:36.027: RADIUS: Received from id 181 172.29.51.235:1646, Accounting-response,
len 20
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 10.1.4.1, we want 10.1.4.1
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Processing AV service=ppp
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Processing AV addr=10.1.4.1
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Processing AV netmask*255.255.255.248
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Processing AV route=10.1.4.0 255.255.255.248
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Authorization succeeded
Jul 25 15:37:36.055: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 10.1.4.1, we want 10.1.4.1
Jul 25 15:37:36.059: AAA: parse name=Virtual-Access1 idb type=21 tty=-1
Jul 25 15:37:36.059: AAA: name=Virtual-Access1 flags=0x11 type=6 shelf=0 slot=0 adapter=0
port=1 channel=0
Jul 25 15:37:36.059: AAA/MEMORY: create_user (0x6165132C) user='2cardpppoa1' ruser=''
port='' rem_addr='' authen_type=CHAP service=PPP priv=1
Jul 25 15:37:36.059: RADIUS: ustruct sharecount=1
Jul 25 15:37:36.063: RADIUS: Initial Transmit  id 182 172.29.51.235:1646,
Accounting-Request, len 91
Jul 25 15:37:36.063:         Attribute 4 6 81010106
Jul 25 15:37:36.063:         Attribute 5 6 00000001
Jul 25 15:37:36.063:         Attribute 61 6 00000005
Jul 25 15:37:36.063:         Attribute 1 13 32636172
Jul 25 15:37:36.063:         Attribute 40 6 00000001
Jul 25 15:37:36.063:         Attribute 6 6 00000002
Jul 25 15:37:36.063:         Attribute 44 10 30303030
Jul 25 15:37:36.063:         Attribute 7 6 00000001
Jul 25 15:37:36.063:         Attribute 8 6 81010401
Jul 25 15:37:36.063:         Attribute 41 6 00000000
Jul 25 15:37:36.119: RADIUS: Received from id 182 172.29.51.235:1646, Accounting-response,
len 20
```

```
Jul 25 15:37:36.119:  SSG: free_user (0x6165132C) user='2cardpppoa1' ruser=''
port='1633984376' rem_addr='' authen_type=CHAP service=PPP priv=1
radius_servertype='0x0000' account_info='0x611BA7BC'
Jul 25 15:37:36.119: AAA/MEMORY: free_user (0x6165132C) user='2cardpppoa1' ruser=''
port='' rem_addr='' authen_type=CHAP service=PPP priv=1
Jul 25 15:37:36.975: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp1mid#
```

SSG Completes the Remote User's Authentication (Step 3 and Step 4[b])

If SSG cannot complete the authentication, use the following **debug** commands in privileged EXEC mode:

- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).
- **debug vtemplate**—Displays cloning information for a virtual access interface from the time it is cloned from a virtual template to the time the virtual access interface comes down when the call ends.

Example 3-25 provides a sample of the **debug** command output that results from these commands.

Example 3-25 Sample Debug SSG Completes the Remote User's Authentication

```
nrp1mid# debug radius

Radius protocol debugging is on

nrp1mid# debug vtemp

nrp1mid# debug vtemplate

Virtual Template debugging is on
nrp1mid#

nrp1mid# sh debug

VTEMPLATE:
  Virtual Template debugging is on
Radius protocol debugging is on
nrp1mid#
Jul 25 15:04:21.970: RADIUS: ustruct sharecount=1
Jul 25 15:04:21.970: RADIUS: Initial Transmit Virtual-Access1 id 149 172.29.51.235:1645,
Access-Request, len 82
Jul 25 15:04:21.970:         Attribute 4 6 81010106
Jul 25 15:04:21.970:         Attribute 5 6 00000001
Jul 25 15:04:21.970:         Attribute 61 6 00000005
Jul 25 15:04:21.970:         Attribute 1 13 32636172
Jul 25 15:04:21.970:         Attribute 3 19 08D5DC82
Jul 25 15:04:21.970:         Attribute 6 6 00000002
Jul 25 15:04:21.970:         Attribute 7 6 00000001
Jul 25 15:04:21.982: RADIUS: Received from id 149 172.29.51.235:1645, Access-Accept, len
50
Jul 25 15:04:21.982:         Attribute 6 6 00000002
Jul 25 15:04:21.982:         Attribute 7 6 00000001
Jul 25 15:04:21.982:         Attribute 8 6 81010401
Jul 25 15:04:21.982:         Attribute 9 6 FFFFFFF8
Jul 25 15:04:21.982:         Attribute 12 6 000005DC
Jul 25 15:04:21.982: RADIUS: Authorize IP address 10.10.20.1
Jul 25 15:04:21.982: RADIUS: Framed-IP-Netmask 255.255.255.248
Jul 25 15:04:21.982: RADIUS: framed-route 10.1.4.0 255.255.255.248
Jul 25 15:04:21.982: RADIUS: ustruct sharecount=3
Jul 25 15:04:21.982: RADIUS: Initial Transmit Virtual-Access1 id 150 172.29.51.235:1646,
Accounting-Request, len 85
Jul 25 15:04:21.982:         Attribute 4 6 81010106
Jul 25 15:04:21.982:         Attribute 5 6 00000001
Jul 25 15:04:21.982:         Attribute 61 6 00000005
Jul 25 15:04:21.986:         Attribute 1 13 32636172
Jul 25 15:04:21.986:         Attribute 40 6 00000001
Jul 25 15:04:21.986:         Attribute 6 6 00000002
Jul 25 15:04:21.986:         Attribute 44 10 30303030
Jul 25 15:04:21.986:         Attribute 7 6 00000001
Jul 25 15:04:21.986:         Attribute 41 6 00000000
Jul 25 15:04:22.050: RADIUS: ustruct sharecount=1
Jul 25 15:04:22.054: RADIUS: Initial Transmit  id 151 172.29.51.235:1646,
Accounting-Request, len 91
Jul 25 15:04:22.054:         Attribute 4 6 81010106
Jul 25 15:04:22.054:         Attribute 5 6 00000001
Jul 25 15:04:22.054:         Attribute 61 6 00000005
Jul 25 15:04:22.054:         Attribute 1 13 32636172
Jul 25 15:04:22.054:         Attribute 40 6 00000001
Jul 25 15:04:22.054:         Attribute 6 6 00000002
Jul 25 15:04:22.054:         Attribute 44 10 30303030
Jul 25 15:04:22.054:         Attribute 7 6 00000001
Jul 25 15:04:22.054:         Attribute 8 6 81010401
Jul 25 15:04:22.054:         Attribute 41 6 00000000
Jul 25 15:04:22.070: RADIUS: Received from id 150 172.29.51.235:1646, Accounting-response,
len 20
Jul 25 15:04:22.154: RADIUS: Received from id 151 172.29.51.235:1646, Accounting-response,
len 20
Jul 25 15:04:22.982: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp1mid#
```

SSG Obtains an IP Address (Step 5)

If SSG cannot obtain an IP address, use the following **debug** command in privileged EXEC mode:

- **debug ip peer**—Displays address activity and contains additional output when pool groups are defined.

Example 3-26 provides a sample of the **debug** command output that results from this command.

Example 3-26 Sample Debug SSG Obtains an IP Address

```
nrp1mid# debug ip peer

IP peer address activity debugging is on
nrp1mid#

nrp1mid# sh debug

Generic IP:
  IP peer address activity debugging is onnrp1mid#
nrp1mid#
Jul 25 15:07:22.032: set_ip_peer_addr: Vi1: address = 10.10.20.1 (4)
Jul 25 15:07:22.072: set_ip_peer_addr: Vi1: address = 10.10.20.1 (4)
Jul 25 15:07:22.076: Vi1 IPCP: Install route to 10.1.4.1
Jul 25 15:07:22.992: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
```

SSG User Browses SSD and Authenticates (Step 6 through Step 10)

The remote user accesses the Service Selection Dashboard (SSD) using a web client pointing to SSD; the user is authenticated. SSG creates a host object for the remote user. Once authenticated, the web page is updated to reflect the services the user is authorized to access and initially only obtains default service, if defined.

If the SSG user cannot browse SSD and authenticate, use the following **debug** commands in privileged EXEC mode:

- **debug ssg events**—Displays event messages for system modules.
- **debug ssg ctrl-events**—Displays all event messages for control modules.
- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).

Example 3-27 provides a sample of the **debug** command output that results from the **debug ssg events** and **debug ssg ctrl-events** commands. Example 3-28 provides a sample of the **debug** command output that results from the **debug radius** command.

Example 3-27 Sample Debug SSG User Browses SSD and Authenticates—ssg events and ssg ctrl-events

```
nrp1mid# debug ssg events

SSG events debugging is on

nrp1mid# debug ssg ctrl-events

SSG control path events debugging is on

nrp1mid# sh debug

SSG:
  SSG control path events debugging is on
  SSG events debugging is on
nrp1mid#
Jul 25 15:12:48.002: SSG-CTL-EVN: Received cmd (1,ssg_user1) from 10.1.4.2.
Jul 25 15:12:48.002: SSG-CTL-EVN: Add cmd=1 from 10.1.4.2 into SSG control cmd queue.
Jul 25 15:12:48.002: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:12:48.002: SSG-CTL-EVN: Handling account logon for host 10.1.4.2.
Jul 25 15:12:48.002: SSG-CTL-EVN: Authenticating user ssg_user1.
Jul 25 15:12:48.026: SSG-CTL-EVN: Creating HostObject for host 10.1.4.2.
Jul 25 15:12:48.026: SSG-EVN: HostObject::HostObject: size = 288

Jul 25 15:12:48.026: SSG-EVN: HostObject::Reset

Jul 25 15:12:48.026: SSG-CTL-EVN: Set Host Mac Address .
Jul 25 15:12:48.026: SSG-CTL-EVN: ** attr->type = 6
Jul 25 15:12:48.026: SSG-CTL-EVN: ATTR_LOOP = 1
Jul 25 15:12:48.026: SSG-CTL-EVN: HostObject::InsertServiceList MP3
Jul 25 15:12:48.026: SSG-CTL-EVN: ATTR_LOOP = 2
Jul 25 15:12:48.030: SSG-CTL-EVN: HostObject::InsertServiceList Warez
Jul 25 15:12:48.030: SSG-CTL-EVN: ATTR_LOOP = 3
Jul 25 15:12:48.030: SSG-CTL-EVN: Account logon is accepted (10.1.4.2,ssg_user1).
Jul 25 15:12:48.030: SSG-CTL-EVN: Send cmd 1 to host 10.1.4.2. dst=172.29.51.236:34966
Jul 25 15:12:48.030: SSG-CTL-EVN: Activating HostObject for host 10.1.4.2.
Jul 25 15:12:48.550: SSG-CTL-EVN: Received cmd (4, ) from 10.1.4.2.
Jul 25 15:12:48.550: SSG-CTL-EVN: Add cmd=4 from 10.1.4.2 into SSG control cmd queue.
Jul 25 15:12:48.550: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:12:48.550: SSG-CTL-EVN: Handling account status query for host 10.1.4.2.
Jul 25 15:12:48.550: SSG-CTL-EVN: Collecting the account info and ack the query.
Jul 25 15:12:48.550: SSG-CTL-EVN: Send cmd 4 to host 10.1.4.2. dst=172.29.51.236:34965
Jul 25 15:12:53.494: SSG-CTL-EVN: Received cmd (4, ) from 10.1.4.2.
Jul 25 15:12:53.494: SSG-CTL-EVN: Add cmd=4 from 10.1.4.2 into SSG control cmd queue.
Jul 25 15:12:53.494: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:12:53.494: SSG-CTL-EVN: Handling account status query for host 10.1.4.2.
Jul 25 15:12:53.494: SSG-CTL-EVN: Collecting the account info and ack the query.
Jul 25 15:12:53.494: SSG-CTL-EVN: Send cmd 4 to host 10.1.4.2. dst=172.29.51.236:34966
Jul 25 15:12:58.503: SSG-EVN: - SSGTimeout Process: Collect 2 Host IPs for Timeout check.
```

Example 3-28 Sample Debug SSG User Browses SSD and Authenticates—RADIUS

```
nrp1mid# debug radius

Radius protocol debugging is on

nrp1mid# sh debug

Radius protocol debugging is on
nrp1mid#
[1st request when the ssg user is logging in to retrieve the service's menu]
Jul 25 15:18:58.944: RADIUS: ustruct sharecount=1
Jul 25 15:18:58.944: RADIUS: Initial Transmit  id 168 172.29.51.235:1645, Access-Request,
len 61
Jul 25 15:18:58.944:         Attribute 4 6 81010106
Jul 25 15:18:58.944:         Attribute 61 6 00000000
Jul 25 15:18:58.944:         Attribute 1 11 7373675F
Jul 25 15:18:58.944:         Attribute 2 18 8F1A945D
Jul 25 15:18:58.952: RADIUS: Received from id 168 172.29.51.235:1645, Access-Accept, len
52
Jul 25 15:18:58.952:         Attribute 6 6 0000000A
Jul 25 15:18:58.952:         Attribute 26 12 00000009FA064E4D
Jul 25 15:18:58.952:         Attribute 26 14 00000009FA084E57
Jul 25 15:18:58.952: RADIUS: saved authorization data for user 611BA988 at 61776FEC
Jul 25 15:18:58.956: RADIUS: ustruct sharecount=1
Jul 25 15:18:58.956: RADIUS: Initial Transmit  id 169 172.29.51.235:1646, Accounting-Re
quest, len 89
Jul 25 15:18:58.956:         Attribute 4 6 81010106
Jul 25 15:18:58.956:         Attribute 5 6 00000000
Jul 25 15:18:58.956:         Attribute 61 6 00000005
Jul 25 15:18:58.956:         Attribute 1 11 7373675F
Jul 25 15:18:58.956:         Attribute 40 6 00000001
Jul 25 15:18:58.956:         Attribute 6 6 00000002
Jul 25 15:18:58.956:         Attribute 44 10 30303030
Jul 25 15:18:58.956:         Attribute 7 6 00000001
Jul 25 15:18:58.956:         Attribute 8 6 81010402
Jul 25 15:18:58.956:         Attribute 41 6 00000000
Jul 25 15:18:59.032: RADIUS: Received from id 169 172.29.51.235:1646, Accounting-respon
se, len 20
nrp1mid#
nrp1mid#
nrp1mid#
```

SSG User Selects a Service From SSD (Step 11)

After authenticating, the user receives an updated web page with the services they are authorized to access. Once the user selects a service, SSD initiates a request to the SSG with the user name, password, and service name. SSG queries the service provider (SP) radius server and receives the service profile. For "proxy" service types, the SSG queries the Cust RADIUS server to authenticate the remote user. SSG can optionally assign an address to the remote user.

If the SSG user cannot select a service from SSD, use the following **debug** commands in privileged EXEC mode:

- **debug ssg events**—Displays event messages for system modules.
- **debug ssg ctrl-events**—Displays all event messages for control modules.
- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).

Example 3-29 provides a sample of the **debug** command output that results from the **debug ssg events** and **debug ssg ctrl-events** commands. Example 3-30 provides a sample of the output that results from the **debug radius** command.

Example 3-29 Sample Debug SSG User Selects a Service from SSD—ssg events and ssg ctrl-events

```
Jul 25 15:12:59.867: SSG-CTL-EVN: Received cmd (11,MP3) from 10.1.4.2.
Jul 25 15:12:59.867: SSG-CTL-EVN: Add cmd=11 from 10.1.4.2 into SSG control cmd queue.
Jul 25 15:12:59.867: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:12:59.867: SSG-CTL-EVN: Handling service logon for host 10.1.4.2.
Jul 25 15:12:59.867: SSG-CTL-EVN: Locating the HostObject for host 10.1.4.2.
Jul 25 15:12:59.867: SSG-CTL-EVN: Checking maximum service count.
Jul 25 15:12:59.867: SSG-CTL-EVN: Downloading service profile for service MP3.
Jul 25 15:12:59.867: SSG-EVN: DownloadProfile: getting profile for MP3 from AAA

Jul 25 15:12:59.879: SSG-CTL-EVN: Creating ServiceInfo for service MP3.
Jul 25 15:12:59.879: SSG-EVN: - ServiceInfo object already exists

Jul 25 15:12:59.879: SSG-EVN: ServiceInfo::Reset

Jul 25 15:12:59.879: SSG-EVN: DstNet::~DstNet

Jul 25 15:12:59.879: SSG-EVN: DstNet::~DstNet: deleting m_pInclude

Jul 25 15:12:59.879: SSG-EVN: NetSegmentList::~NetSegmentList

Jul 25 15:12:59.879: SSG-EVN: DstNet::DstNet: size = 12

Jul 25 15:12:59.879: SSG-EVN: NetSegmentList::NetSegmentList: size = 4

Jul 25 15:12:59.879: SSG-CTL-EVN: Checking service mode.
Jul 25 15:12:59.879: SSG-CTL-EVN: ServiceLogon: Enqueue request of service MP3
Jul 25 15:12:59.879: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:12:59.879: SSG-CTL-EVN: Handling service logon for host 10.1.4.2.
Jul 25 15:12:59.879: SSG-CTL-EVN: Locating the HostObject for host 10.1.4.2.
Jul 25 15:12:59.879: SSG-CTL-EVN: Creating ConnectionObject (10.1.4.2, MP3).
Jul 25 15:12:59.879: SSG-EVN: ConnectionObject::ConnectionObject: size = 112

Jul 25 15:12:59.879: SSG-EVN: Opening connection for user ssg_user1

Jul 25 15:12:59.879: SSG-CTL-EVN: Service logon is accepted.
Jul 25 15:12:59.879: SSG-CTL-EVN: Send cmd 11 to host 10.1.4.2. dst=172.29.51.236:34965
Jul 25 15:12:59.879: SSG-CTL-EVN: Activating the ConnectionObject.
Jul 25 15:12:59.879: SSG-CTL-EVN:  Acct Start: local: user_name=ssg_user1

Jul 25 15:12:59.879: SSG-CTL-EVN: GetConnClass NULL len 0
Jul 25 15:12:59.879: SSG-CTL-EVN: Get Host Mac Address 0000.0000.0000
Jul 25 15:12:59.879: SSG-CTL-EVN: SSG: Accounting:: AddCiscoVSA
Jul 25 15:12:59.879: SSG-CTL-EVN:  SSG: Accounting:: AddCiscoVSA add serviceName=MP3
Jul 25 15:12:59.883: SSG-CTL-EVN:  AddCiscoVSA: add serviceUserName=ssg_user1

Jul 25 15:12:59.883: SSG-CTL-EVN:  AddCiscoVSA: no adding V attr  pService 611BBE78

Jul 25 15:13:00.335: SSG-CTL-EVN: Received cmd (4, ) from 10.1.4.2.
Jul 25 15:13:00.335: SSG-CTL-EVN: Add cmd=4 from 10.1.4.2 into SSG control cmd queue.
Jul 25 15:13:00.335: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:13:00.335: SSG-CTL-EVN: Handling account status query for host 10.1.4.2.
Jul 25 15:13:00.335: SSG-CTL-EVN: Collecting the account info and ack the query.
Jul 25 15:13:00.335: SSG-CTL-EVN: Send cmd 4 to host 10.1.4.2. dst=172.29.51.236:34966
Jul 25 15:13:00.383: SSG-CTL-EVN: Received cmd (4,MP3) from 10.1.4.2.
Jul 25 15:13:00.383: SSG-CTL-EVN: Add cmd=4 from 10.1.4.2 into SSG control cmd queue.
Jul 25 15:13:00.383: SSG-CTL-EVN: Dequeue a cmd from the cmd queue and pass it to cmd
handler.
Jul 25 15:13:00.383: SSG-CTL-EVN: Handling account status query for host 10.1.4.2.
Jul 25 15:13:00.383: SSG-CTL-EVN: Collecting the account info and ack the query.
Jul 25 15:13:00.383: SSG-CTL-EVN: Send cmd 4 to host 10.1.4.2. dst=172.29.51.236:34965
nrp1mid#
```

Example 3-30 Sample Debug SSG User Selects a Service From SSD—RADIUS

```
[2nd request when user has selected the MP3 service. A radius request is sent to obtain
the attributes of that service profile]

nrp1mid#
Jul 25 15:19:29.062: RADIUS: ustruct sharecount=1
Jul 25 15:19:29.062: RADIUS: Initial Transmit  id 170 172.29.51.235:1645, Access-Reques
t, len 61
Jul 25 15:19:29.062:         Attribute 4 6 81010106
Jul 25 15:19:29.062:         Attribute 61 6 00000000
Jul 25 15:19:29.066:         Attribute 1 5 4D503302
Jul 25 15:19:29.066:         Attribute 2 18 31D0AD9B
Jul 25 15:19:29.066:         Attribute 6 6 00000005
Jul 25 15:19:29.070: RADIUS: Received from id 170 172.29.51.235:1645, Access-Accept, le
n 95
Jul 25 15:19:29.074:         Attribute 6 6 00000005
Jul 25 15:19:29.074:         Attribute 26 19 00000009FB0D494D
Jul 25 15:19:29.074:         Attribute 26 30 00000009FB185231
Jul 25 15:19:29.074:         Attribute 26 10 00000009FB044D43
Jul 25 15:19:29.074:         Attribute 26 10 00000009FB045450
Jul 25 15:19:29.074: RADIUS: saved authorization data for user 611BA988 at 61776FEC
Jul 25 15:19:29.074: RADIUS: ustruct sharecount=1
Jul 25 15:19:29.074: RADIUS: Initial Transmit  id 171 172.29.51.235:1646, Accounting-Re
quest, len 89
Jul 25 15:19:29.074:         Attribute 4 6 81010106
Jul 25 15:19:29.074:         Attribute 5 6 00000000
Jul 25 15:19:29.074:         Attribute 61 6 00000005
Jul 25 15:19:29.074:         Attribute 1 11 7373675F
Jul 25 15:19:29.074:         Attribute 40 6 00000001
Jul 25 15:19:29.074:         Attribute 6 6 00000002
Jul 25 15:19:29.074:         Attribute 44 10 30303030
Jul 25 15:19:29.074:         Attribute 7 6 00000001
Jul 25 15:19:29.074:         Attribute 8 6 81010402
Jul 25 15:19:29.074:         Attribute 41 6 00000000
Jul 25 15:19:29.162: RADIUS: Received from id 171 172.29.51.235:1646, Accounting-response,
len 20
nrp1mid#
nrp1mid#
```

Verifying Correct Configuration for PPPoX Remote Access SSG to MPLS VPN Integration

The following **show** commands are useful in debugging PPPoX Remote Access SSG to MPLS VPN integration:

- **show atm pvc ppp**—Displays all ATM permanent virtual circuits (PVCs) and traffic information; applies only to PPPoA.
- **show int virtual-access** *virtual access interface #*—Displays status, traffic data, and configuration information about a specified virtual access interface.
- **show vpdn session** [**all**]—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network (VPDN); PPPoE specific.
- **show vpdn tunnel**—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network (VPDN); PPPoE specific.
- **show ssg host** [**host address**]—Displays the information about a subscriber and current connections of the subscriber.
- **show ssg binding**—Displays service names that have been bound to interfaces and the IP addresses to which they have been bound.
- **show ssg connection** *host ip service*—Displays the connections of a given host and a service name.
- **show ssg service** *service*—Displays the information for a service.

Example 3-31 shows the detailed output that results when you implement these **show** commands.

Example 3-31 Sample show Command Output for PPPoX Remote Access SSG to MPLS VPN Integration

```
nrp1mid# sh atm pvc ppp

           VCD /                                           Peak Avg/Min Burst
ATM Int.   Name       VPI    VCI  Type      VA VASt  SC     Kbps Kbps    Cells VCSt
0/0/0.3033 3           30     33   PVC       1   UP UBR   155000                UP

nrp1mid# sh int virtual-access 1

Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback0 (10.1.1.6)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Bound to ATM0/0/0.3033 VCD: 3, VPI: 30, VCI: 33
  Cloned from virtual-template: 1
  Last input 00:00:00, output never, output hang never
  Last clearing of "show interface" counters 18:41:07
  Queueing strategy: fifo
  Output queue 0/40, 0 drops; input queue 0/75, 0 drops
  5 minute input rate 1000 bits/sec, 3 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     45630 packets input, 2288960 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
     20723 packets output, 815083 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions



nrp1bot# sh vpdn session all

%No active L2TP tunnels

%No active L2F tunnels

%No active PPTP tunnels

PPPoE Session Information Total tunnels 1 sessions 1

session id: 1
local MAC address: 0002.b992.7807, remote MAC address: 00a0.ccd9.2697
virtual access interface: Vi2, outgoing interface: AT0/0/0, vc: 21/33
    16 packets sent, 161 received
    258 bytes sent, 8887 received

nrp1bot# sh vpdn tunnel

%No active L2TP tunnels

%No active L2F tunnels

%No active PPTP tunnels

PPPoE Tunnel Information Total tunnels 1 sessions 1

PPPoE Tunnel Information

Session count: 1
nrp1bot#

nrp1mid# sh ssg host

1: 10.1.4.1
2: 10.1.4.2

nrp1mid# sh ssg host 10.1.4.2

------------------------ HostObject Content -----------------------
Activated: TRUE
Interface:
User Name: ssg_user1
Host IP: 10.1.4.2
Msg IP: 172.29.51.236 (9902)
Host DNS IP: 0.0.0.0
Maximum Session Timeout: 0 seconds
Host Idle Timeout: 0 seconds
Class Attr: NONE
User logged on since: 16:13:46.000 UTC Wed Jul 25 2001
User last activity at: 16:13:51.000 UTC Wed Jul 25 2001
```

```
Default Service: NONE
DNS Default Service: NONE
Active Services: NONE
AutoService: NONE

Subscribed Services: MP3; Warez;



nrp1mid# sh ssg bindi

        Warez                 -> ATM0/0/0.201
        MP3                   -> ATM0/0/0.200


nrp1mid# sh ssg connection 10.1.4.2 MP3

------------------------ ConnectionObject Content -----------------------
User Name: ssg_user1
Owner Host: 10.1.4.2
Associated Service: MP3
Connection State: 0 (UP)
Connection Started since:
15:19:29.000 UTC Wed Jul 25 2001
User last activity at: 15:23:29.000 UTC Wed Jul 25 2001
Connection Traffic Statistics:
        Input Bytes = 9680 (HI = 0), Input packets = 242
        Output Bytes = 9680 (HI = 0), Output packets = 242

nrp1mid# sh ssg service MP3

------------------------ ServiceInfo Content -----------------------
Uplink IDB: ATM0/0/0.200
Name: MP3
Type: PASS-THROUGH
Mode: CONCURRENT
Service Session Timeout: 0 seconds
Service Idle Timeout: 0 seconds
Authentication Type: CHAP

DNS Server(s):

Included Network Segments:
        10.21.0.0/255.255.0.0
Excluded Network Segments:
ConnectionCount 1
Full User Name not used


Domain List:

Active Connections:
        1   : RealIP=0.0.0.0, Subscriber=10.1.4.2
```

L2TP to MPLS VPN Integration

This section contains the following troubleshooting topics:

- Overview of DSL L2TP to MPLS VPN Integration, page 3-41
- Initiating and Viewing debug Command Output, page 3-42

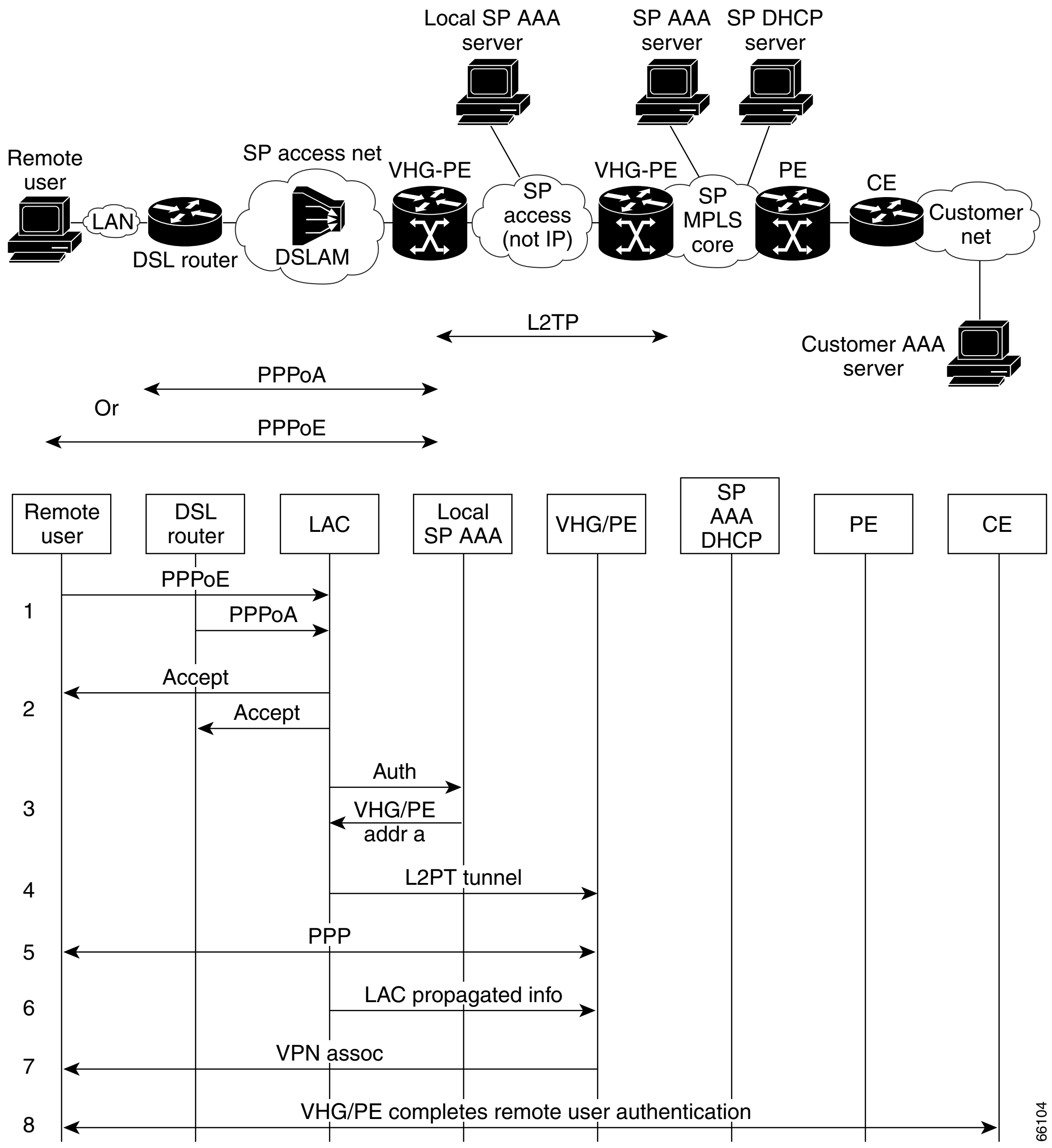
- Debugging Problems Associated with DSL L2TP to MPLS VPN Integration, page 3-43, including:
  - Accept Connection (Step 2), page 3-43
  - Authenticate and Obtain Tunnel Information (Step 3), page 3-44
  - Establish L2TP Tunnel (Step 4), page 3-46
  - Create Tunnel Session and Propagate PPP Information (Step 5 through Step 10[a]), page 3-48
  - Complete PPP Authentication (Step 5 through Step 10[b]), page 3-49
- Verifying Correct Configuration for DSL L2TP to MPLS VPN Integration, page 3-51, including:
  - show Commands for NAS, page 3-51
  - show Commands for VHG-PE/LNS, page 3-53

Overview of DSL L2TP to MPLS VPN Integration

Figure 3-5 shows the overall topology of an integrated DSL to MPLS VPN solution, within the context of a VPN-capable service provider's MPLS backbone. In this scenario, the customer has outsourced all remote access operations to its service provider. In addition, but not explicitly shown in Figure 3-5, the service provider operates an MPLS VPN that interconnects all customer sites. An incoming PPPoX session (which arrives at the NAS 64000 NRP) is L2TP-tunneled to the VHG/PE 6400 NRP, which maps the session to the corresponding VRF.

Note DSL L2TP to MPLS VPN integration is very similar to the dial-in L2TP to MPLS VPN solution discussed in the "Troubleshooting Dial Access to MPLS VPN Integration" section on page 2-1.

Figure 3-5 DSL L2TP



Initiating and Viewing debug Command Output

For reminders on using the command line interface for viewing **debug** command output, refer to the sections of the *Remote Access to MPLS VPN Solution Provisioning Guide 1.0* entitled “User Interface Command Modes” and “User Command Modes.” This document can be accessed at the following URL:

http://cco/univercd/cc/td/doc/product/vpn/solution/rampls/index.htm

Debugging Problems Associated with DSL L2TP to MPLS VPN Integration

Table 3-5 (below) corresponds to Figure 3-5, providing a cross-reference to the troubleshooting topics associated with DSL L2TP to MPLS VPN integration.

Table 3-5 Troubleshooting Topics for DSL L2TP to MPLS VPN Integration

Event Shown in Figure 3-5:	Related Troubleshooting Topics:
Line 1: User initiates a PPPoE session or the DSL router initiates a PPPoA session.	
Line 2: NAS accepts the PPPoX session.	Accept Connection (Step 2), page 3-43
Line 3: NAS partially authenticates the user with CHAP or PAP and obtains tunnel information.	Authenticate and Obtain Tunnel Information (Step 3), page 3-44
Line 4: NAS initiates a tunnel to the VHG-PE/LNS (if L2TP tunnel does not exist).	Establish L2TP Tunnel (Step 4), page 3-46
Lines 5-10: PPP session is created and connection is extended to terminate on the VHG-PE/LNS; NAS propagates available PPP information.	Create Tunnel Session and Propagate PPP Information (Step 5 through Step 10[a]), page 3-48
Lines 5-10: VHG-PE/LNS completes the authentication, associates the remote user with a specific customer MPLS VPN, and obtains an IP address.	Complete PPP Authentication (Step 5 through Step 10[b]), page 3-49

Accept Connection (Step 2)

If NAS does not accept the PPPoX session, use the following **debug** commands in privileged EXEC mode:

- **debug ppp negotiation**—Displays PPP packets transmitted during PPP startup, where PPP options are negotiated.
- **debug ppp authentication**—Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.
- **debug vpdn pppoe-events**—Displays PPPoE protocol errors that prevent a session from being established or errors that cause an established session to be closed.

Example 3-32 provides a sample of the **debug** command output that results from these commands.

Example 3-32 Sample Debug Accept Connection

```
[LAC Only]

nrp1mid# debug ppp negotiation

PPP protocol negotiation debugging is on

nrp1mid# debug ppp authentication

PPP authentication debugging is on

nrp1mid# sh debug
```

```
PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on
Jul 26 17:49:19.120: Vi1 LCP: I CONFREQ [Listen] id 114 len 14
Jul 26 17:49:19.120: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul 26 17:49:19.120: Vi1 LCP:    MRU 2048 (0x01040800)
Jul 26 17:49:19.120: Vi1 LCP: O CONFREQ [Listen] id 175 len 15
Jul 26 17:49:19.120: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul 26 17:49:19.120: Vi1 LCP:    MagicNumber 0x07BEC6D3 (0x050607BEC6D3)
Jul 26 17:49:19.120: Vi1 LCP: O CONFACK [Listen] id 114 len 14
Jul 26 17:49:19.120: Vi1 LCP:    MagicNumber 0xA60C0000 (0x0506A60C0000)
Jul 26 17:49:19.120: Vi1 LCP:    MRU 2048 (0x01040800)
Jul 26 17:49:19.156: Vi1 LCP: I CONFACK [ACKsent] id 175 len 15
Jul 26 17:49:19.156: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul 26 17:49:19.156: Vi1 LCP:    MagicNumber 0x07BEC6D3 (0x050607BEC6D3)
Jul 26 17:49:19.156: Vi1 LCP: State is Open
Jul 26 17:49:19.156: Vi1 PPP: Phase is AUTHENTICATING, by this end [0 sess, 0 load]
Jul 26 17:49:19.156: Vi1 CHAP: O CHALLENGE id 27 len 28 from "nrp1mid"
Jul 26 17:49:19.192: Vi1 CHAP: I RESPONSE id 27 len 39 from "anchan@gcoe.com"
Jul 26 17:49:19.192: Vi1 PPP: Phase is FORWARDING [0 sess, 0 load]
Jul 26 17:49:20.208: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
```

Authenticate and Obtain Tunnel Information (Step 3)

If NAS does not authenticate or obtain tunnel information, use the following **debug** commands in privileged EXEC mode:

- **debug vpdn events**—Displays messages about events that are part of normal tunnel establishment or shutdown.
- **debug aaa authentication**—Displays information on AAA/Terminal Access Controller Access Control System Plus (TACACS+) authentication.
- **debug aaa authorization**—Displays information on AAA/TACACS+ authorization. To disable debugging output, use the **no** form of this command.
- **debug radius**—Displays information associated with the Remote Authentication Dial-In User Server (RADIUS).

Example 3-33 provides a sample of the **debug** command output that results from these commands.

Example 3-33 Sample Debug Authenticate and Obtain Tunnel Information

```
nrp1mid# debug vpdn event

VPDN events debugging is on

nrp1mid# sh debug

VPN:
  VPDN events debugging is on
nrp1mid#
Jul 26 17:59:14.202: VPDN: Domain based tunnel lookup. Domain is gcoe.com
Jul 26 17:59:14.202: Vi1 VPDN: Looking for tunnel -- gcoe.com --
Jul 26 17:59:14.230: Vi1 VPDN/RPMS/: Got tunnel info for gcoe.com

[LAC Only]

nrp1mid# debug aaa author

AAA Authorization debugging is on
```

```
nrp1mid# debug aaa authentication

AAA Authentication debugging is on

nrp1mid# debug radius

Radius protocol debugging is on

nrp1mid# sh debug

General OS:
  AAA Authentication debugging is on
  AAA Authorization debugging is on

Radius protocol debugging is on
nrp1mid#
Jul 26 17:55:17.204: AAA: parse name=Virtual-Access1 idb type=21 tty=-1
Jul 26 17:55:17.204: AAA: name=Virtual-Access1 flags=0x11 type=6 shelf=0 slot=0 adapter=0
port=1 channel=0
Jul 26 17:55:17.204: AAA/MEMORY: create_user (0x61622070) user='gcoe.com' ruser=''
port='Virtual-Access1' rem_addr='' authen_type=NONE service=LOGIN priv=0
Jul 26 17:55:17.204: Virtual-Access1 AAA/AUTHOR/VPDN (3484971203): Port='Virtual-Access1'
list='default' service=NET
Jul 26 17:55:17.204: AAA/AUTHOR/VPDN: Virtual-Access1 (3484971203) user='gcoe.com'
Jul 26 17:55:17.204: Virtual-Access1 AAA/AUTHOR/VPDN (3484971203): send AV service=ppp
Jul 26 17:55:17.204: Virtual-Access1 AAA/AUTHOR/VPDN (3484971203): send AV protocol=vpdn
Jul 26 17:55:17.204: Virtual-Access1 AAA/AUTHOR/VPDN (3484971203): found list "default"
Jul 26 17:55:17.204: Virtual-Access1 AAA/AUTHOR/VPDN (3484971203): Method=radius (radius)
Jul 26 17:55:17.204: RADIUS: authenticating to get author data
Jul 26 17:55:17.204: RADIUS: ustruct sharecount=2
Jul 26 17:55:17.204: RADIUS: Initial Transmit Virtual-Access1 id 215 172.29.51.235:1645,
Access-Request, len 75
Jul 26 17:55:17.204:         Attribute 4 6 81010106
Jul 26 17:55:17.204:         Attribute 5 6 00000001
Jul 26 17:55:17.204:         Attribute 61 6 00000005
Jul 26 17:55:17.204:         Attribute 1 13 72656462
Jul 26 17:55:17.204:         Attribute 2 18 C0859715
Jul 26 17:55:17.204:         Attribute 6 6 00000005
Jul 26 17:55:17.216: RADIUS: Received from id 215 172.29.51.235:1645, Access-Accept, len
87
Jul 26 17:55:17.216:         Attribute 6 6 00000005
Jul 26 17:55:17.216:         Attribute 64 6 01000003
Jul 26 17:55:17.216:         Attribute 66 12 01313239
Jul 26 17:55:17.216:         Attribute 67 12 01313239
Jul 26 17:55:17.216:         Attribute 69 21 013BEAC4
Jul 26 17:55:17.216:         Attribute 90 10 016E7270
Jul 26 17:55:17.216: RADIUS: saved authorization data for user 61622070 at 6167E678
Jul 26 17:55:17.216: RADIUS: Tunnel-Type, [01] 00 00 03
Jul 26 17:55:17.216: RADIUS: TAS(1) created and enqueued.
Jul 26 17:55:17.216: RADIUS: Tunnel-Client-Endpoint, [01] 10.1.1.6
Jul 26 17:55:17.216: RADIUS: Tunnel-Server-Endpoint, [01] 10.1.1.7
Jul 26 17:55:17.216: RADIUS: Tunnel-Password decrypted, [01] bodega
Jul 26 17:55:17.216: RADIUS: Tunnel-Cli-Auth, [01] nrp1mid
Jul 26 17:55:17.216: RADIUS: TAS(1) takes precedence over tagged attributes,
tunnel_type=l2tp
Jul 26 17:55:17.216: RADIUS: Tunnel-Medium-Type not set. Use default type 'IP'.
Jul 26 17:55:17.216: RADIUS: Tunnel-Medium-Type not set. Use default type 'IP'.
Jul 26 17:55:17.216: RADIUS: free TAS(1)
Jul 26 17:55:17.216: AAA/AUTHOR (3484971203): Post authorization status = PASS_REPL
Jul 26 17:55:17.216: AAA/AUTHOR/VPDN: Processing AV tunnel-type*l2tp
Jul 26 17:55:17.216: AAA/AUTHOR/VPDN: Processing AV ip-addresses=10.1.1.7
Jul 26 17:55:17.216: AAA/AUTHOR/VPDN: Processing AV source-ip=10.1.1.6
Jul 26 17:55:17.216: AAA/AUTHOR/VPDN: Processing AV tunnel-id=nrp1mid
Jul 26 17:55:17.216: AAA/AUTHOR/VPDN: Processing AV l2tp-tunnel-password=bodega
Jul 26 17:55:17.216: AAA/AUTHOR/VPDN: Processing AV tunnel-tag*1
Jul 26 17:55:17.216: AAA/MEMORY: free_user (0x61622070) user='gcoe.com' ruser=''
port='Virtual-Access1' rem_addr='' authen_type=NONE service=LOGIN priv=0
Jul 26 17:55:17.220: AAA: parse name=Virtual-Access1 idb type=21 tty=-1
Jul 26 17:55:17.220: AAA: name=Virtual-Access1 flags=0x11 type=6 shelf=0 slot=0 adapter=0
port=1 channel=0
Jul 26 17:55:17.224: AAA/MEMORY: create_user (0x61622070) user='anchan@gcoe.com' ruser=''
port='Virtual-Access1' rem_addr='' authen_type=CHAP service=PPP priv=1
Jul 26 17:55:17.224: RADIUS: ustruct sharecount=3
Jul 26 17:55:17.224: RADIUS: Initial Transmit Virtual-Access1 id 216 172.29.51.235:1646,
Accounting-Request, len 159
Jul 26 17:55:17.228:         Attribute 4 6 81010106
Jul 26 17:55:17.228:         Attribute 5 6 00000001
Jul 26 17:55:17.228:         Attribute 61 6 00000005
Jul 26 17:55:17.228:         Attribute 1 20 616E6368
Jul 26 17:55:17.228:         Attribute 40 6 00000001
Jul 26 17:55:17.228:         Attribute 45 6 00000002
Jul 26 17:55:17.228:         Attribute 6 6 00000002
Jul 26 17:55:17.228:         Attribute 44 10 30303030
Jul 26 17:55:17.228:         Attribute 7 6 00000001
Jul 26 17:55:17.228:         Attribute 67 13 01313239
Jul 26 17:55:17.228:         Attribute 66 13 01313239
Jul 26 17:55:17.228:         Attribute 90 11 016E7270
Jul 26 17:55:17.228:         Attribute 91 11 016E7270
Jul 26 17:55:17.228:         Attribute 68 13 32303937
Jul 26 17:55:17.228:         Attribute 41 6 00000000
Jul 26 17:55:17.268: RADIUS: Received from id 216 172.29.51.235:1646, Accounting-response,
len 20
Jul 26 17:55:18.224: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp1mid#
```

Establish L2TP Tunnel (Step 4)

If the NAS cannot establish a tunnel, use the following **debug** commands in privileged EXEC mode:

- **debug vpdn event**—Displays messages about events that are part of normal tunnel establishment or shutdown.
- **debug vpdn 12x-events**—Displays errors associated with L2X protocol events.

Example 3-34 provides a sample of the **debug** command output that results from these commands.

Example 3-34 Sample Debug Establish L2TP Tunnel

```
[LAC Only]

nrp1mid# debug vpdn event

VPDN events debugging is on

nrp1mid# debug vpdn l2x-events

L2X protocol events debugging is on

nrp1mid# sh debug

VPN:
  L2X protocol events debugging is on
  VPDN events debugging is on
nrp1mid#
```

```
nrp1mid#
nrp1mid#
Jul 26 18:04:15.198: VPDN: Domain based tunnel lookup. Domain is gcoe.com
Jul 26 18:04:15.198: Vi1 VPDN: Looking for tunnel -- gcoe.com --
Jul 26 18:04:15.222: Vi1 VPDN/RPMS/: Got tunnel info for gcoe.com
Jul 26 18:04:15.222: Vi1 VPDN/RPMS/:   LAC nrp1mid
Jul 26 18:04:15.222: Vi1 VPDN/RPMS/:   source-ip 10.1.1.6
Jul 26 18:04:15.222: Vi1 VPDN/RPMS/:   l2tp-busy-disconnect yes
Jul 26 18:04:15.222: Vi1 VPDN/RPMS/:   l2tp-tunnel-password xxxxxx
Jul 26 18:04:15.222: Vi1 VPDN/RPMS/:   IP 10.1.1.7
Jul 26 18:04:15.222: Vi1 VPDN/: curlvl 1 Address 0: 10.1.1.7,  priority 1
Jul 26 18:04:15.222: Vi1 VPDN/: Select non-active address 10.1.1.7, priority 1
Jul 26 18:04:15.222: Tnl 41197 L2TP: SM State idle
Jul 26 18:04:15.222: Tnl 41197 L2TP: O SCCRQ
Jul 26 18:04:15.226: Tnl 41197 L2TP: Tunnel state change from idle to wait-ctl-reply
Jul 26 18:04:15.226: Tnl 41197 L2TP: SM State wait-ctl-reply
Jul 26 18:04:15.226: Vi1 VPDN: Find LNS process created
Jul 26 18:04:15.226: Vi1 VPDN: Forward to address 10.1.1.7
Jul 26 18:04:15.226: Vi1 VPDN: Pending
Jul 26 18:04:15.226: Vi1 VPDN: Process created
Jul 26 18:04:15.226: Tnl 41197 L2TP: I SCCRP from nrp1mid
Jul 26 18:04:15.226: Tnl 41197 L2TP: Got a challenge from remote peer, nrp1mid
Jul 26 18:04:15.226: Tnl 41197 L2TP: Got a response from remote peer, nrp1mid
Jul 26 18:04:15.226: Tnl 41197 L2TP: Tunnel Authentication success
Jul 26 18:04:15.226: Tnl 41197 L2TP: Tunnel state change from wait-ctl-reply to
established
Jul 26 18:04:15.226: Tnl 41197 L2TP: O SCCCN  to nrp1mid tnlid 29847
Jul 26 18:04:15.230: Tnl 41197 L2TP: SM State established
Jul 26 18:04:15.230: Vi1 VPDN: Forwarding...
Jul 26 18:04:15.230: Vi1 VPDN: Bind interface direction=1
Jul 26 18:04:15.230: Tnl/Cl 41197/28 L2TP: Session FS enabled
Jul 26 18:04:15.230: Tnl/Cl 41197/28 L2TP: Session state change from idle to
wait-for-tunnel
Jul 26 18:04:15.230: Vi1 Tnl/Cl 41197/28 L2TP: Create session
Jul 26 18:04:15.230: Tnl 41197 L2TP: SM State established
Jul 26 18:04:15.230: Vi1 Tnl/Cl 41197/28 L2TP: O ICRQ to nrp1mid 29847/0
Jul 26 18:04:15.230: Vi1 Tnl/Cl 41197/28 L2TP: Session state change from wait-for-tunnel
to wait-reply
Jul 26 18:04:15.230: Vi1 VPDN: anchan@gcoe.com is forwarded
Jul 26 18:04:15.234: Vi1 Tnl/Cl 41197/28 L2TP: O ICCN to nrp1mid 29847/28
Jul 26 18:04:15.234: Vi1 Tnl/Cl 41197/28 L2TP: Session state change from wait-reply to
established
Jul 26 18:04:16.230: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp1mid#

[LNS Only]

nrp2mid# debug vpdn event

VPDN events debugging is on

nrp2mid# debug vpdn l2x-events

L2X protocol events debugging is on

nrp2mid# sh debug

VPN:
  L2X protocol events debugging is on
  VPDN events debugging is on
nrp2mid#
Jul 26 18:04:15.226: L2TP: I SCCRQ from nrp1mid tnl 41197
Jul 26 18:04:15.226: Tnl 29847 L2TP: Got a challenge in SCCRQ, nrp1mid
Jul 26 18:04:15.226: Tnl 29847 L2TP: New tunnel created for remote nrp1mid, address
10.1.1.6
Jul 26 18:04:15.226: Tnl 29847 L2TP: O SCCRP  to nrp1mid tnlid 41197
Jul 26 18:04:15.226: Tnl 29847 L2TP: Tunnel state change from idle to wait-ctl-reply
Jul 26 18:04:15.230: Tnl 29847 L2TP: I SCCCN from nrp1mid tnl 41197
Jul 26 18:04:15.230: Tnl 29847 L2TP: Got a Challenge Response in SCCCN from nrp1mid
Jul 26 18:04:15.230: Tnl 29847 L2TP: Tunnel Authentication success
Jul 26 18:04:15.230: Tnl 29847 L2TP: Tunnel state change from wait-ctl-reply to
established
Jul 26 18:04:15.230: Tnl 29847 L2TP: SM State established
Jul 26 18:04:15.230: Tnl 29847 L2TP: I ICRQ from nrp1mid tnl 41197
Jul 26 18:04:15.230: Tnl/Cl 29847/28 L2TP: Session FS enabled
Jul 26 18:04:15.230: Tnl/Cl 29847/28 L2TP: Session state change from idle to wait-connect
Jul 26 18:04:15.234: Tnl/Cl 29847/28 L2TP: New session created
Jul 26 18:04:15.234: Tnl/Cl 29847/28 L2TP: O ICRP to nrp1mid 41197/28
Jul 26 18:04:15.234: Tnl/Cl 29847/28 L2TP: I ICCN from nrp1mid tnl 41197, cl 28
Jul 26 18:04:15.234: Tnl/Cl 29847/28 L2TP: Session state change from wait-connect to
established
Jul 26 18:04:15.234: Vi1 VPDN: Virtual interface created for anchan@gcoe.com
Jul 26 18:04:15.234: Vi1 VPDN: Clone from Vtemplate 1 filterPPP=0 blocking
Jul 26 18:04:15.258: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
Jul 26 18:04:15.258: Vi1 VPDN: Bind interface direction=2
Jul 26 18:04:15.258: Vi1 VPDN: PPP LCP accepted rcv CONFACK
Jul 26 18:04:15.258: Vi1 VPDN: PPP LCP accepted sent CONFACK
Jul 26 18:04:16.270: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp2mid#
```

Create Tunnel Session and Propagate PPP Information (Step 5 through Step 10[a])

If the NAS and the VHG/PE cannot authenticate each other, use the following **debug** commands in privileged EXEC mode:

- **debug ppp authentication**—Displays authentication protocol messages, including Challenge Authentication Protocol (CHAP) packet exchanges and Password Authentication Protocol (PAP) exchanges.
- **debug ppp negotiation**—Displays PPP packets transmitted during PPP startup, where PPP options are negotiated.

Example 3-35 provides a sample of the **debug** command output that results from these commands.

Example 3-35 Sample Debug Create Tunnel Session and Propagate PPP Information

```
[LNS Only]

nrp2mid# debug ppp authentication

PPP authentication debugging is on

nrp2mid# debug ppp negotiation

PPP protocol negotiation debugging is on

nrp2mid# sh debug

PPP:
  PPP authentication debugging is on
  PPP protocol negotiation debugging is on

nrp2mid#
Jul 26 18:10:34.197: Vi1 PPP: Phase is DOWN, Setup [0 sess, 0 load]
Jul 26 18:10:34.221: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
Jul 26 18:10:34.221: Vi1 PPP: Using set call direction
Jul 26 18:10:34.221: Vi1 PPP: Treating connection as a callin
Jul 26 18:10:34.221: Vi1 PPP: Phase is ESTABLISHING, Passive Open [0 sess, 0 load]
Jul 26 18:10:34.221: Vi1 PPP: No remote authentication for call-in
Jul 26 18:10:34.221: Vi1 LCP: State is Listen
Jul 26 18:10:34.221: Vi1 LCP: I FORCED CONFREQ len 11
Jul 26 18:10:34.221: Vi1 LCP:    AuthProto CHAP (0x0305C22305)
Jul 26 18:10:34.221: Vi1 LCP:    MagicNumber 0x07D22332 (0x050607D22332)
Jul 26 18:10:34.221: Vi1 PPP: Phase is AUTHENTICATING, by this end [0 sess, 0 load]
Jul 26 18:10:34.221: Vi1 CHAP: O CHALLENGE id 31 len 28 from "nrp2mid"
Jul 26 18:10:34.221: Vi1 CHAP: I RESPONSE id 31 len 39 from "anchan@gcoe.com"
Jul 26 18:10:34.233: Vi1 CHAP: O SUCCESS id 31 len 4
Jul 26 18:10:34.249: Vi1 PPP: Phase is UP [0 sess, 0 load]
Jul 26 18:10:34.249: Vi1 IPCP: O CONFREQ [Closed] id 1 len 10
Jul 26 18:10:34.249: Vi1 IPCP:    Address 10.60.1.1 (0x03060A3C0101)
Jul 26 18:10:34.289: Vi1 IPCP: I CONFREQ [REQsent] id 210 len 10
Jul 26 18:10:34.289: Vi1 IPCP:    Address 0.0.0.0 (0x030600000000)
Jul 26 18:10:34.289: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 0.0.0.0, we want 0.0.0.0
Jul 26 18:10:34.289: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 0.0.0.0, we want 0.0.0.0
Jul 26 18:10:34.289: Vi1 IPCP: Pool returned 10.60.1.2
Jul 26 18:10:34.289: Vi1 IPCP: O CONFNAK [REQsent] id 210 len 10
Jul 26 18:10:34.289: Vi1 IPCP:    Address 10.60.1.2 (0x03060A3C0102)
Jul 26 18:10:34.293: Vi1 IPCP: I CONFACK [REQsent] id 1 len 10
Jul 26 18:10:34.293: Vi1 IPCP:    Address 10.60.1.1 (0x03060A3C0101)
Jul 26 18:10:34.329: Vi1 IPCP: I CONFREQ [ACKrcvd] id 211 len 10
Jul 26 18:10:34.329: Vi1 IPCP:    Address 10.60.1.2 (0x03060A3C0102)
Jul 26 18:10:34.329: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 10.60.1.2, we want 10.60.1.2
Jul 26 18:10:34.329: Vi1 AAA/AUTHOR/IPCP: Reject 10.60.1.2, using 10.60.1.2
Jul 26 18:10:34.329: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 10.60.1.2, we want 10.60.1.2
Jul 26 18:10:34.329: Vi1 IPCP: O CONFACK [ACKrcvd] id 211 len 10
Jul 26 18:10:34.329: Vi1 IPCP:    Address 10.60.1.2 (0x03060A3C0102)
Jul 26 18:10:34.329: Vi1 IPCP: State is Open
Jul 26 18:10:34.333: Vi1 IPCP: Install route to 10.60.1.2
Jul 26 18:10:35.241: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
Jul 26 18:10:36.221: Vi1 LCP: TIMEout: State Open
Jul 26 18:10:44.113: Vi1 LCP: I ECHOREQ [Open] id 212 len 8 magic 0xA60C0000
Jul 26 18:10:44.113: Vi1 LCP: O ECHOREP [Open] id 212 len 8 magic 0x07D22332
Jul 26 18:10:54.110: Vi1 LCP: I ECHOREQ [Open] id 213 len 8 magic 0xA60C0000
Jul 26 18:10:54.110: Vi1 LCP: O ECHOREP [Open] id 213 len 8 magic 0x07D22332
Jul 26 18:10:54.194: Vi1 LCP: I ECHOREP [Open] id 1 len 12 magic 0xA60C0000
Jul 26 18:10:54.194: Vi1 LCP: Received id 1, sent id 1, line up
nrp2mid#
```

Complete PPP Authentication (Step 5 through Step 10[b])

If the tunnel is not created and the remaining PPP information is not propagated, use the following **debug** commands in privileged EXEC mode:

- **debug aaa per-user**—Displays information about the per-user configuration downloaded from the AAA server.
- **debug vtemplate**—Displays cloning information for a virtual access interface from the time it is cloned from a virtual template to the time the virtual access interface comes down when the call ends.
- **debug ip peer**—Displays address activity and contains additional output when pool groups are defined.

Example 3-36 provides a sample of the **debug** command output that results from these commands.

Example 3-36 Sample Debug Complete PPP Authentication

```
[LNS Only]

nrp2mid# debug aaa per-user

AAA Per-user attributes debugging is on

nrp2mid# debug vtemplate

Virtual Template debugging is on

nrp2mid# debug ip peer

IP peer address activity debugging is on

nrp2mid# sh debug

General OS:
  AAA Per-user attributes debugging is on

Generic IP:
  IP peer address activity debugging is on
VTEMPLATE:
  Virtual Template debugging is on
nrp2mid#
Jul 26 18:13:39.212: Vt1 VTEMPLATE: Unable to create and clone vaccess
Jul 26 18:13:39.212: Vi1 VTEMPLATE: Reuse Vi1, recycle queue size 0
Jul 26 18:13:39.212: Vi1 VTEMPLATE: Hardware address 0002.b992.7793
Jul 26 18:13:39.212: Vi1 VTEMPLATE: Has a new cloneblk vtemplate, now it has vtemplate
Jul 26 18:13:39.212: Vi1 VTEMPLATE: ************* CLONE VACCESS1 *****************
Jul 26 18:13:39.212: Vi1 VTEMPLATE: Clone from Virtual-Template1
interface Virtual-Access1
default ip address
no ip address
encap ppp
no ip address
ip mroute-cache
end

Jul 26 18:13:39.236: %LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
Jul 26 18:13:39.248: Vi1 VTEMPLATE: Has a new cloneblk AAA, now it has vtemplate/AAA
Jul 26 18:13:39.248: Vi1 VTEMPLATE: ************* CLONE VACCESS1 *****************
Jul 26 18:13:39.248: Vi1 VTEMPLATE: Clone from AAA
interface Virtual-Access1
 ip vrf forwarding vpndsl
 ip unnumbered loopback100
 peer default ip address pool vpndslpool
end

Jul 26 18:13:39.304: Vi1: Pools to search : vpndslpool
Jul 26 18:13:39.304: Vi1: Pool vpndslpool returned address = 10.60.1.2
Jul 26 18:13:39.344: Vi1 AAA/AUTHOR/PER-USER: Event IP_UP
Jul 26 18:13:39.344: Vi1 AAA/AUTHOR: IP_UP
Jul 26 18:13:39.344: Vi1 AAA/PER-USER: processing author params.
Jul 26 18:13:39.344: Vi1 IPCP: Install route to 10.60.1.2
Jul 26 18:13:40.244: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1,
changed state to up
nrp2mid#
```

Verifying Correct Configuration for DSL L2TP to MPLS VPN Integration

This section provides the following examples of how you can show information for the events outlined in:

- show Commands for NAS, page 3-51
- show Commands for VHG-PE/LNS, page 3-53

The information provided here applies only to DSL access to MPLS VPN integration. For more information about the configuration and troubleshooting tasks associated with L2TP, please refer to the Cisco *Configuring Virtual Private Networks* document at the following URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/dialns_c/dnsprt3/dcdvpn.htm

show Commands for NAS

To verify the details of L2TP tunnel setup and session on the NAS, use the following **show** commands:

- **show vpdn**—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network.
- **show vpdn tunnel**—Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.
- **show vpdn tunnel all**—Displays VPN tunnel information including tunnel protocol, ID, local and remote tunnel names, packets sent and received, tunnel, and transport status.
- **show vpdn session**—Displays VPN session information including interface, tunnel, username, packets, status, and window statistics.
- **show user**—Displays information about the active lines on the router per a specific user.
- **show caller**—Displays individual users and consumed resources on the NAS/LAC, and active call statistics for large pools of connections, and the absolute and idle times for each user.
- **show caller user** *caller name*—Displays the **show caller** command information for a particular user.
- **show vpdn history failure**—Displays the content of the failure history table for the user.
- **show resource-pool resource**—RPMS specific; displays resource groups configured in the network access server.

Example 3-37 provides a sample of the **debug** command output that results from these commands.

Example 3-37 Sample show Command Output for NAS

```
[LAC Only]

nrp1mid# sh vpdn

L2TP Tunnel and Session Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
48083 43245 nrp1mid       est    10.1.1.1 1701   1

LocID RemID TunID Intf       Username      State  Last Chg Fastswitch
30    30    48083 Vi1        anchan@gcoe est    00:00:46 enabled

%No active L2F tunnels

%No active PPTP tunnels
```

```
%No active PPPoE tunnels

[LNS Only]

nrp2mid# sh vpdn

L2TP Tunnel and Session Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
43245 48083 nrp1mid       est    10.1.1.6       1701  1

LocID RemID TunID Intf       Username      State  Last Chg Fastswitch
30    30    43245 Vi1        anchan@gcoe est    00:00:50 enabled

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels

nrp1mid# sh vpdn tunnel

L2TP Tunnel Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
48083 43245 nrp1mid       est    10.1.1.7       1701  1

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels


nrp1mid# sh vpdn tunnel all

L2TP Tunnel Information Total tunnels 1 sessions 1

Tunnel id 48083 is up, remote id is 43245, 1 active sessions
  Tunnel state is established, time since change 00:03:44
  Remote tunnel name is nrp1mid
    Internet Address 10.1.1.7, port 1701
  Local tunnel name is nrp1mid
    Internet Address 10.1.1.6, port 1701
  53 packets sent, 47 received
  4222 bytes sent, 1026 received
  Control Ns 4, Nr 2
  Local RWS 4500 (default), Remote RWS 4500 (max)
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 0, max 2
  Total resends 0, ZLB ACKs sent 0
  Current nosession queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0
  Sessions disconnected due to lack of resources 0

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels
nrp1mid#
```

```
nrp1mid# sh vpdn session

L2TP Session Information Total tunnels 1 sessions 1

LocID RemID TunID Intf       Username      State  Last Chg Fastswitch
30    30    48083 Vi1        anchan@gcoe est    00:04:39 enabled

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels


nrp1mid# sh user
    Line       User       Host(s)              Idle       Location
*  2 vty 0                idle                 00:00:00 10.21.65.35
  Vi1          anchan@gcoe Virtual PPP (ATM   ) 00:06:34

  Interface  User      Mode                     Idle Peer Address

nrp1mid# sh caller
                                                Active    Idle
  Line         User               Service       Time      Time
  vty 0        -                  VTY           00:35:25  00:00:00
  Vi1          anchan@gcoe.com PPP   ATM     00:07:38  00:07:39


nrp1mid# sh caller user anchan@gcoe.com

  User: anchan@gcoe.com, line Vi1, service PPP ATM
        Active time 00:07:48, Idle time 00:07:49
  Timeouts:            Absolute  Idle
      Limits:          -         -
      Disconnect in:   -         -
  PPP: LCP Open, CHAP (<- none)
  IP: Local 10.1.1.6
  VPDN: NAS nrp1mid, MID 30, MID Unknown
        HGW , NAS CLID 0, HGW CLID 0, tunnel open
  Counts: 23313 packets input, 914960 bytes, 0 no buffer
          0 input errors, 0 CRC, 0 frame, 0 overrun
          17392 packets output, 214903 bytes, 0 underruns
          0 output errors, 0 collisions, 0 interface resets

nrp1mid#

nrp1mid# sh vpdn history failure

Table size: 20
Number of entries in table: 1

User: anchan@gcoe.com, MID = 29
NAS: nrp1mid, IP address = 10.1.1.6, CLID = 0
Gateway: Information is not applicable
Log time: 23:47:32, Error repeat count: 26
Failure type: The remote server closed this session
Failure reason: Result 2, Error 6, Disconnect from PPP
```

show Commands for VHG-PE/LNS

To verify the details of the L2TP tunnel setup, PPP sessions, virtual access interface configurations, and local address pool assignment on the VHG-PE/LNS, use the following **show** commands:

- **show vpdn**—Displays information about active Level 2 Forwarding (L2F) protocol tunnel and message identifiers in a Virtual Private Dialup Network.
- **show user**—Displays information about the active lines on the router per a specific user.
- **show caller**—Displays individual users and consumed resources on the NAS/LAC, and active call statistics for large pools of connections, and the absolute and idle times for each user.
- **show caller user** *caller name*—Displays the "show caller" information for a particular user.
- **show interface virtual-access** *virtual-access #*—Displays detailed information about the virtual access interface.
- **show virtual access configuration**—Displays detailed information about the virtual access interface configuration.
- **show ip route** [**vrf** *vrf name*]—Displays the current state of the routing table, including the IP address, the network mask, protocol, and static routes.
- **show ip local pool**—Displays the address pools that have been downloaded to a Cisco network access server.
- **show vpdn history failure**—Displays the content of the failure history table for the user.

Example 3-38 shows the detailed output that would result from these show commands.

Example 3-38 Sample show Command Output for VHG-PE/LNS

```
nrp2mid# sh vpdn tunnel

L2TP Tunnel Information Total tunnels 1 sessions 1

LocID RemID Remote Name   State  Remote Address  Port  Sessions
43245 48083 nrp1mid       est    10.1.1.6       1701  1

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels
nrp2mid#



nrp2mid# sh vpdn tunnel all

L2TP Tunnel Information Total tunnels 1 sessions 1

Tunnel id 43245 is up, remote id is 48083, 1 active sessions
  Tunnel state is established, time since change 00:03:49
  Remote tunnel name is nrp1mid
    Internet Address 10.1.1.6, port 1701
  Local tunnel name is nrp1mid
    Internet Address 10.1.2.26, port 1701
  49 packets sent, 56 received
  1070 bytes sent, 2386 received
  Control Ns 2, Nr 4
  Local RWS 4500 (default), Remote RWS 4500 (max)
  Retransmission time 1, max 1 seconds
  Unsent queuesize 0, max 0
  Resend queuesize 0, max 1
  Total resends 0, ZLB ACKs sent 2
  Current nosession queue check 0 of 5
  Retransmit time distribution: 0 0 0 0 0 0 0 0 0
  Sessions disconnected due to lack of resources 0

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels
nrp2mid#

nrp2mid# sh vpdn  session

L2TP Session Information Total tunnels 1 sessions 1

LocID RemID TunID Intf       Username      State  Last Chg Fastswitch
30    30    43245 Vi1        anchan@gcoe est    00:04:45 enabled

%No active L2F tunnels

%No active PPTP tunnels

%No active PPPoE tunnels
nrp2mid#



nrp2mid# sh user

    Line       User       Host(s)              Idle       Location
*  2 vty 0                idle                 00:00:00 10.21.65.35
   3 vty 1                idle                 23:47:18 ssd
  Vi1          anchan@gcoe Virtual PPP (L2TP  ) 00:00:01

  Interface  User      Mode                    Idle Peer Address


nrp1mid# sh caller
                                                Active    Idle
  Line         User               Service       Time      Time
  vty 0        -                  VTY           00:35:25  00:00:00
  Vi1          anchan@gcoe.com PPP   ATM     00:07:38  00:07:39

nrp2mid# sh caller user anchan@gcoe.com

  User: anchan@gcoe.com, line Vi1, service PPP L2TP
        Active time 00:08:04, Idle time 00:00:00
  Timeouts:            Absolute  Idle
      Limits:          -         -
      Disconnect in:   -         -
  PPP: LCP Open, CHAP (<- AAA), IPCP
  IP: Local 10.60.1.1, remote 10.60.1.2
  VPDN: NAS nrp1mid, MID 30, MID Unknown
        HGW nrp1mid, NAS CLID 0, HGW CLID 0, tunnel open
  Counts: 117 packets input, 3752 bytes, 0 no buffer
          0 input errors, 0 CRC, 0 frame, 0 overrun
          103 packets output, 1446 bytes, 0 underruns
          0 output errors, 0 collisions, 0 interface resets

nrp2mid# sh vpdn history failure

Table size: 20
Number of entries in table: 1

User: anchan@gcoe.com, MID = 22
NAS: nrp1mid, IP address = 10.1.1.6, CLID = 0
```

```
Gateway: nrp1mid, IP address = 10.1.2.26, CLID = 0
Log time: 22:49:53, Error repeat count: 18
Failure type: The remote server closed this session
Failure reason: Result 2, Error 6, Disconnect from PPP

nrp2mid# sh int virtual-access 1

Virtual-Access1 is up, line protocol is up
  Hardware is Virtual Access interface
  Interface is unnumbered. Using address of Loopback100 (10.60.1.1)
  MTU 1500 bytes, BW 100000 Kbit, DLY 100000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
  DTR is pulsed for 5 seconds on reset
  LCP Open
  Open: IPCP
  Last input 00:00:02, output never, output hang never
  Last clearing of "show interface" counters 01:42:43
  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
     1399 packets input, 48172 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
     1231 packets output, 17238 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions

nrp2mid# sh int virtual-access 1 configuration

Virtual-Access1 is an L2TP link interface

Building configuration...

interface Virtual-Access1 configuration...
ip vrf forwarding vpndsl
ip unnumbered Loopback100
ip mtu 1460
ip mroute-cache

nrp2mid# sh ip route vrf vpndsl

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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/32 is subnetted, 2 subnets
C       10.60.1.2 is directly connected, Virtual-Access1
C       10.60.1.1 is directly connected, Loopback100
nrp2mid#

nrp2mid# sh ip local pool

 Pool                     Begin          End             Free  In use
 vpn-nrp                  6.0.0.1        6.0.0.254        254       0
 vpndslpool             10.60.1.2       10.60.1.5          3       1
```

Troubleshooting Specific Features

The features described here may be used with various dsl access methods. This section describes how to troubleshoot the feature itself. For information on troubleshooting the main call flow for the specific access method, see the access method sections.

This section includes:

Troubleshooting the Framed-Route VRF Aware Feature, page 3-57

Verifying and Troubleshooting On-demand Address Pools, page 3-61

Troubleshooting the Framed-Route VRF Aware Feature

On the VHG/PE, verify that the subnet sent to the CPE is in the appropriate VRF routing table:

show ip route vrf <vrf name>

If the subnet is not in the correct VRF routing table, troubleshoot the RADIUS exchange between the VHG/PE and the RADIUS AR server, checking to make sure the AV pair containing the subnet is being exchanged. Use the following commands:

debug aaa authorization

debug aaa authentication

debug aaa per-user

debug radius

debug ip routing vrf *vrf name to which PPP session belongs*

Example 3-39 Example of VHG/PE show ip route Command Output

```
c72d9-1#
*Sep  4 09:42:33.627: AAA/AUTHOR (0x55): Pick method list 'default'
*Sep  4 09:42:33.631: AAA/AUTHEN/PPP (00000055): Pick method list 'default'
*Sep  4 09:42:33.631: RADIUS: Pick NAS IP for uid=85 tableid=0 cfg_addr=10.10.104.9
best_addr=0.0.0.0
*Sep  4 09:42:33.631: RADIUS/ENCODE(00000055): acct_session_id: 146
*Sep  4 09:42:33.631: RADIUS(00000055): sending
*Sep  4 09:42:33.631: RADIUS(00000055): Send to unknown id 21647/157 10.10.100.3:1645,
Access-Request, len 103
*Sep  4 09:42:33.635: RADIUS:  authenticator 96 9E 2F 52 E4 9E 98 10 - E5 B1 B4 77 F5 F4
40 63
*Sep  4 09:42:33.635: RADIUS:  Framed-Protocol     [7]   6   PPP                      [1]
*Sep  4 09:42:33.635: RADIUS:  User-Name           [1]   24  "U0001N1P3V1.9@V1.9.com"
*Sep  4 09:42:33.635: RADIUS:  CHAP-Password       [3]   19  *
*Sep  4 09:42:33.635: RADIUS:  NAS-Port-Type       [61]  6   ISDN                     [2]
*Sep  4 09:42:33.635: RADIUS:  Called-Station-Id   [30]  6   "9111"
*Sep  4 09:42:33.635: RADIUS:  Service-Type        [6]   6   Framed                   [2]
*Sep  4 09:42:33.635: RADIUS:  NAS-IP-Address      [4]   6   10.10.104.9
*Sep  4 09:42:33.635: RADIUS:  Acct-Session-Id     [44]  10  "00000092"
*Sep  4 09:42:33.639: RADIUS: Received from id 21647/157 10.10.100.3:1645, Access-Accept,
len 478
*Sep  4 09:42:33.639: RADIUS:  authenticator AA 76 9F 6E 15 06 14 5D - 4B DA F0 6C E6 25
D3 C4
*Sep  4 09:42:33.639: RADIUS:  Service-Type        [6]   6   Framed                   [2]
```

```
*Sep  4 09:42:33.639: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
*Sep  4 09:42:33.639: RADIUS:  Vendor, Cisco       [26]  83
*Sep  4 09:42:33.639: RADIUS:   Cisco AVpair       [1]   77  "lcp:interface-config=ip vrf
forwarding V1.9.com \n ip unnumbered loopback 9"
*Sep  4 09:42:33.639: RADIUS:  Vendor, Cisco       [26]  75
*Sep  4 09:42:33.639: RADIUS:   Cisco AVpair       [1]   69  "ip:route=vrf V1.9.com
192.168.200.0 255.255.255.0 32.1.9.10 tag 250"
*Sep  4 09:42:33.639: RADIUS(00000055): Received from id 21647/157
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: service-type
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: Framed-Protocol
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: interface-config:Peruser I/F
*Sep  4 09:42:33.643: ppp118 PPP/AAA: Check Attr: route:Peruser
*Sep  4 09:42:33.663: %LINK-3-UPDOWN: Interface Virtual-Access10, changed state to up
*Sep  4 09:42:33.663: AAA/AUTHEN/PPP (00000055): Pick method list 'default'
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: Process Author
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: Process Attr: interface-config
*Sep  4 09:42:33.663: AAA/AUTHOR: Processing PerUser AV interface-config
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: Process Attr: interface-config
*Sep  4 09:42:33.663: Vi10 AAA/AUTHOR/LCP: IF_config:
ip vrf forwarding V1.9.com \n ip unnumbered loopback 9

*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: FSM authorization not needed
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/FSM: We can start IPCP
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Start.  Her address 32.1.9.10, we want 0.0.0.0
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Reject 32.1.9.10, using 0.0.0.0
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Processing AV route
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Authorization succeeded
*Sep  4 09:42:33.719: Vi10 AAA/AUTHOR/IPCP: Done.  Her address 32.1.9.10, we want 0.0.0.0
*Sep  4 09:42:33.727: AAA/AUTHOR: Processing PerUser AV route
*Sep  4 09:42:33.727: Vi10 AAA/PERUSER/ROUTE: route string: IP route vrf V1.9.com
192.168.200.0 255.255.255.0 32.1.9.10 tag 250

*Sep  4 09:42:33.735: RT(V1.9.com): closer admin distance for 32.1.9.10, flushing 1 routes
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED queued, Queue size 1
*Sep  4 09:42:33.735: RT(V1.9.com): add 32.1.9.10/32 via 0.0.0.0, connected metric [0/0]
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED push
*Sep  4 09:42:33.735: RT(V1.9.com): NET-RED queued, Queue size 2
*Sep  4 09:42:33.747: AAA/PER-USER: command = [IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250
]
*Sep  4 09:42:33.747: AAA/PER-USER: line = [IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250]
*Sep  4 09:42:33.751: RT(V1.9.com): add 192.168.200.0/24 via 32.1.9.10, static metric
[1/0]
*Sep  4 09:42:33.751: RT(V1.9.com): NET-RED 192.168.200.0/24
*Sep  4 09:42:33.751: RT(V1.9.com): NET-RED queued, Queue size 1
*Sep  4 09:42:33.763: is_up: 1 state: 4 sub state: 1 line: 0 has_route: True
*Sep  4 09:42:34.663: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access10,
changed state to up
```

When you disconnect, you will see the static route being removed:

```
*Sep  4 09:56:43.713: %LINK-3-UPDOWN: Interface Virtual-Access10, changed state to down
*Sep  4 09:56:43.713: is_up: 0 state: 0 sub state: 1 line: 0 has_route: True
*Sep  4 09:56:43.713: RT(V1.9.com): interface Virtual-Access10 removed from routing table
*Sep  4 09:56:43.713: RT(V1.9.com): Pruning routes for Virtual-Access10 (1)
*Sep  4 09:56:43.713: RT(V1.9.com): delete route to 32.1.9.10 via 0.0.0.0,
Virtual-Access10
*Sep  4 09:56:43.713: RT(V1.9.com): no routes to 32.1.9.10, flushing
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED 32.1.9.10/32
```

```
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED queued, Queue size 1
*Sep  4 09:56:43.713: RT(V1.9.com): add 32.1.9.10/32 via 0.0.0.0, static metric [1/0]
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED 32.1.9.10/32
*Sep  4 09:56:43.713: RT(V1.9.com): NET-RED queued, Queue size 2
*Sep  4 09:56:44.713: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access10,
changed state to down
c72d9-1#
c72d9-1#
*Sep  4 09:57:03.712: AAA/PER-USER: command = [no IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250]
*Sep  4 09:57:03.712: AAA/PER-USER: line = [no IP route vrf V1.9.com 192.168.200.0
255.255.255.0 32.1.9.10 tag 250]
*Sep  4 09:57:03.724: AAA/AUTHOR: decrement ref cnt for ip route 192.168.200.0
255.255.255.0 32.1.9.10 to 0
*Sep  4 09:57:03.724: RT(V1.9.com): del 192.168.200.0 via 32.1.9.10, static metric [1/0]
*Sep  4 09:57:03.724: RT(V1.9.com): delete network route to 192.168.200.0
*Sep  4 09:57:03.724: RT(V1.9.com): NET-RED 192.168.200.0/24
*Sep  4 09:57:03.724: RT(V1.9.com): NET-RED queued, Queue size 1


Show ip route output:

c72d9-1#sh ip rout vrf V1.9.com conn
     32.0.0.0/32 is subnetted, 2 subnets
C       32.1.9.10 is directly connected, Virtual-Access10
C       32.1.9.241 is directly connected, Loopback9

c72d9-1#sh ip route vrf V1.9.com stat
U    192.168.200.0/24 [1/0] via 32.1.9.10

 V1.9.com is the VRf to which the PPP session belongs
 U means  per-user static route ( a route downloaded via AAA)
```

Example 3-40 Example of RADIUS debug Command Output

```
RADIUS server# show debug
General OS:
  AAA Authentication debugging is on
  AAA Authorization debugging is on
AAA Per-user attributes debugging is on
Radius protocol debugging is on
Radius packet protocol debugging is on

03:01:48: AAA/AUTHEN/PPP (00000001): Pick method list 'default'
03:01:48: RADIUS/ENCODE(00000001): acct_session_id: 23
03:01:48: RADIUS(00000001): sending
03:01:48: RADIUS: Send to unknown id 21 40.40.40.40:1645, Access-Request, len
12
2
03:01:48: RADIUS:  authenticator 28 70 0F 52 08 58 19 B4 - BE 26 71 53 D1 61 17

C6
03:01:48: RADIUS:  User-Name           [1]   16  "827@FRtest.com"
03:01:48: RADIUS:  CHAP-Password       [3]   19  *
03:01:48: RADIUS:  NAS-Port            [5]   6   1
03:01:48: RADIUS:  Vendor, Cisco       [26]  33
03:01:48: RADIUS:   Cisco AVpair       [1]   27  "interface=Virtual-Access1"
03:01:48: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
03:01:48: RADIUS:  Service-Type        [6]   6   Framed                    [2]
03:01:48: RADIUS:  NAS-IP-Address      [4]   6   40.100.100.1
03:01:48: RADIUS:  Acct-Session-Id     [44]  10  "00000017"
03:01:48: RADIUS: Received from id 21 40.40.40.40:1645, Access-Accept, len 165
```

```
03:01:48: RADIUS:  authenticator 1E D4 FB 0D 2D 7C 0E 0A - 6A 4D 83 11 DD
4D 50
AF
03:01:48: RADIUS:  Service-Type        [6]   6   Framed                    [2]
03:01:48: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
03:01:48: RADIUS:  Framed-IP-Address   [8]   6   40.8.8.1
03:01:48: RADIUS:  Framed-IP-Netmask   [9]   6   255.255.255.224
03:01:48: RADIUS:  Framed-Routing      [10]  6   0
03:01:48: RADIUS:  Vendor, Cisco       [26]  60
03:01:48: RADIUS:   Cisco AVpair       [1]   54  "lcp:interface-config#1= ip vrf
 forwarding FRtest.com"
03:01:48: RADIUS:  Vendor, Cisco       [26]  55
03:01:48: RADIUS:   Cisco AVpair       [1]   49  "lcp:interface-config#2= ip unn
umbered loopback1"
03:01:48: RADIUS: Received from id 1
03:01:48: Vi1 PPP/AAA: Check Attr: service-type
03:01:48: Vi1 PPP/AAA: Check Attr: Framed-Protocol
03:01:48: Vi1 PPP/AAA: Check Attr: addr
03:01:48: Vi1 PPP/AAA: Check Attr: route:Peruser
03:01:48: Vi1 PPP/AAA: Check Attr: netmask
03:01:48: Vi1 PPP/AAA: Check Attr: routing
03:01:48: Vi1 PPP/AAA: Check Attr: interface-config:Peruser I/F
03:01:48: Vi1 PPP/AAA: Check Attr: interface-config:Peruser I/F
03:01:48: Vi1 AAA/AUTHOR/LCP: Process Author
03:01:48: Vi1 AAA/AUTHOR/LCP: Vaccess Required for PER USER attrs
03:01:48: Vi1 AAA/AUTHOR/LCP: Process Attr: interface-config
03:01:48: AAA/AUTHOR: Processing PerUser AV interface-config
03:01:48: Vi1 AAA/AUTHOR/LCP: Process Attr: interface-config
03:01:48: AAA/AUTHOR: Processing PerUser AV interface-config
03:01:48: Vi1 AAA/AUTHOR/LCP: IF_config:
 ip vrf forwarding FRtest.com
 ip unnumbered loopback1

03:01:48: RADIUS/ENCODE(00000001): Unsupported AAA attribute timezone
03:01:48: RADIUS(00000001): sending
03:01:48: RADIUS: Send to unknown id 42 40.40.40.40:1646, Accounting-Request,
le
n 159
03:01:48: RADIUS:  authenticator 29 02 40 0E 83 FE 2B 4A - F0 53 32 DC CF 66
0A
EA
03:01:48: RADIUS:  Vendor, Cisco       [26]  32
03:01:48: RADIUS:   Cisco AVpair       [1]   26  "connect-progress=Call Up"
03:01:48: RADIUS:  User-Name           [1]   16  "827@FRtest.com"
03:01:48: RADIUS:  Acct-Status-Type    [40]  6   Start                     [1]
03:01:48: RADIUS:  Acct-Session-Id     [44]  10  "00000017"
03:01:48: RADIUS:  Framed-Protocol     [7]   6   PPP                       [1]
03:01:48: RADIUS:  Authentic           [45]  6   RADIUS                    [1]
03:01:48: RADIUS:  NAS-Port            [5]   6   1
03:01:48: RADIUS:  Vendor, Cisco       [26]  33
03:01:48: RADIUS:   Cisco AVpair       [1]   27  "interface=Virtual-Access1"
03:01:48: RADIUS:  NAS-Port-Type       [61]  6   Virtual                   [5]
03:01:48: RADIUS:  Service-Type        [6]   6   Framed                    [2]
03:01:48: RADIUS:  NAS-IP-Address      [4]   6   40.100.100.1
03:01:48: RADIUS:  Delay-Time          [41]  6   0
03:01:48: Vi1 AAA/AUTHOR/IPCP: Already authorized
03:01:48: Vi1 AAA/AUTHOR/FSM: We can start IPCP
03:01:48: Vi1 AAA/AUTHOR/IPCP: Start.  Her address 40.8.8.1, we want 0.0.0.0
03:01:48: Vi1 AAA/AUTHOR/IPCP: Processing AV addr
03:01:48: Vi1 AAA/AUTHOR/IPCP: Processing AV route
03:01:48: Vi1 AAA/AUTHOR/IPCP: Processing AV netmask
03:01:48: Vi1 AAA/AUTHOR/IPCP: Processing AV routing
03:01:48: Vi1 AAA/AUTHOR/IPCP: Set routing to FALSE
03:01:48: Vi1 AAA/AUTHOR/IPCP: Authorization succeeded
```

```
03:01:48: Vi1 AAA/AUTHOR/IPCP: Done.  Her address 40.8.8.1, we want 40.8.8.1
03:01:48: Vi1 AAA/AUTHOR/IPCP: no author-info for primary dns
03:01:48: Vi1 AAA/AUTHOR/IPCP: no author-info for primary wins
03:01:48: Vi1 AAA/AUTHOR/IPCP: no author-info for seconday dns
03:01:48: Vi1 AAA/AUTHOR/IPCP: no author-info for seconday wins
03:01:48: RADIUS: Received from id 42 40.40.40.40:1646, Accounting-response, len
 20
03:01:48: RADIUS:  authenticator 2A F5 ED C9 72 03 34 BF - BD D4 BE 04 5F A5
E9
04
03:01:48: AAA/AUTHOR: Processing PerUser AV route
03:01:48: Vi1 AAA/PERUSER/ROUTE: vrf name for vaccess: FRtest.com
03:01:48: Vi1 AAA/PERUSER/ROUTE: route string: ip route vrf FRtest.com
40.8.8.0
255.255.255.224 40.8.8.1
```

Verifying and Troubleshooting On-demand Address Pools

In on-demand address pools (ODAP), a central SP RADIUS server manages a block of addresses for each customer. Each pool is divided into subnets of various sizes, and the server assigns subnets to the VHG/PE or NAS/PE on request.

The VHG/PE or NAS/PE acts as a DHCP server. On the VHG/PE or NAS/PE, one on-demand pool is configured for each customer VPN supported by that router. Upon configuration, the VHG/PE or NAS/PE's pool manager requests an initial subnet from the server.

Address management is on demand because address pool subnets are allocated or released based on a threshold. If use exceeds a defined ceiling threshold, the pool manager requests an additional subnet from the server and adds it to the on-demand pool. If use falls below a floor threshold, the pool manager attempts to free one, or more then one, of the on-demand pool's subnets to return it to the server. The VRF routing table on the VHG/PE or NAS/PE is updated with the subnet route whenever a range of addresses is requested from the AR.

If a problem occurs in ODAP, use the commands shown in Table 3-6 on the VHG/PE or NAS/PE. Example 3-41 shows the results of **show up dhcp pool** and Example 3-42 shows the results of **debug ip dhcp server events**.

Table 3-6 show and debug Commands for ODAP

Command	Use To...
show ip dhcp pool <address pool name>	Check that DHCP pool hands out IP addresses for incoming PPP session and puts it in the correct VRF.
debug ip dhcp server events	Report server events such as address assignments.

Example 3-41 Sample Results of show ip dhcp pool for Troubleshooting ODAP

```
Router# show ip dhcp pool odap-test
Pool odap-test : Utilization mark (high/low)    : 80 / 20 Subnet size (first/next)       :
27 / 27 (autogrow) VRF name                      : V1.1.com Total addresses
: 30 Leased addresses               : 0 Pending requests               : 0 1 subnet is
currently in the pool :Current index        IP address range           Leased addresses
42.1.1.1             42.1.1.1   - 42.1.1.30
```

Example 3-42 Sample Results of debug ip dhcp server events for Troubleshooting ODAP

```
Router# debug ip dhcp server events
DHCPD: allocate request for client U1000N1P3V1.1@V1.1.com on Virtual-Access7.
DHCPD: locate VRF V1.1.com pool odap-test for client U1000N1P3V1.1@V1.1.com.
DHCPD: assigned IP address 42.1.1.1 to client
5531.3030.304e.3150.3356.312e.3140.5631.2e31.2e63.6f6d.
```

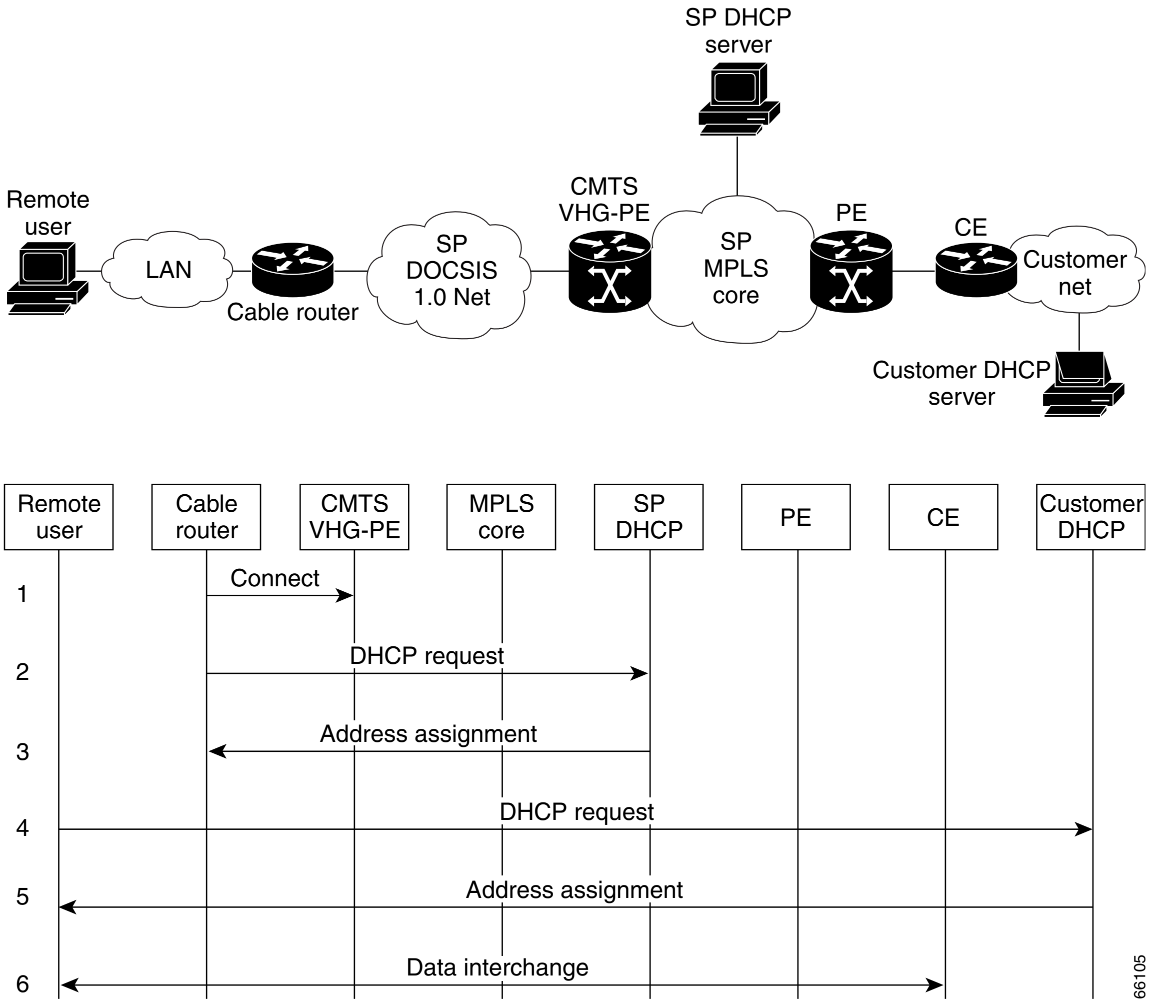
Troubleshooting Cable Access to MPLS VPN Integration

This chapter contains the following information about cable Data Over Cable Service Interface Specification (DOCSIS) 1.0 service identifier (SID) to MPLS VPN integration:

Overview of Cable DOCSIS 1.0 SID to MPLS VPN Integration

Figure 4-1 shows the topology associated with a VPN-capable service provider's MPLS backbone. In this scenario, you should assume that the customer has outsourced all remote access operations to its service provider.

Figure 4-1 Cable DOCSIS 1.0 SID



In DOCSIS 1.0, all traffic from a given cable router (or cable modem) carries the same service identifier (SID). On the Cable Modem Termination System (CMTS) virtual home gateway (VHG)/ provider edge router (PE), all traffic with the same SID value terminates on the same subinterface. At the CMTS VHG/PE, the subinterface is statically configured to map all traffic to a specific VPN Routing and Forwarding (VRF) instance. As a result, traffic from all customer premises equipment (CPE) behind a given cable router is mapped to the same VPN. There is no remote user authorization and authentication in this architecture. Address assignment is based on Dynamic Host Configuration Protocol (DHCP).

Initiating and Viewing debug Command Output

For tips and reminders on using the command- line interface (CLI) for viewing the different debug outputs shown throughout this chapter, refer to "Initiating and Viewing Command Output" section on page 1-2.

Debugging Problems Associated with Cable DOCSIS 1.0 SID to MPLS VPN Integration

Section Overview

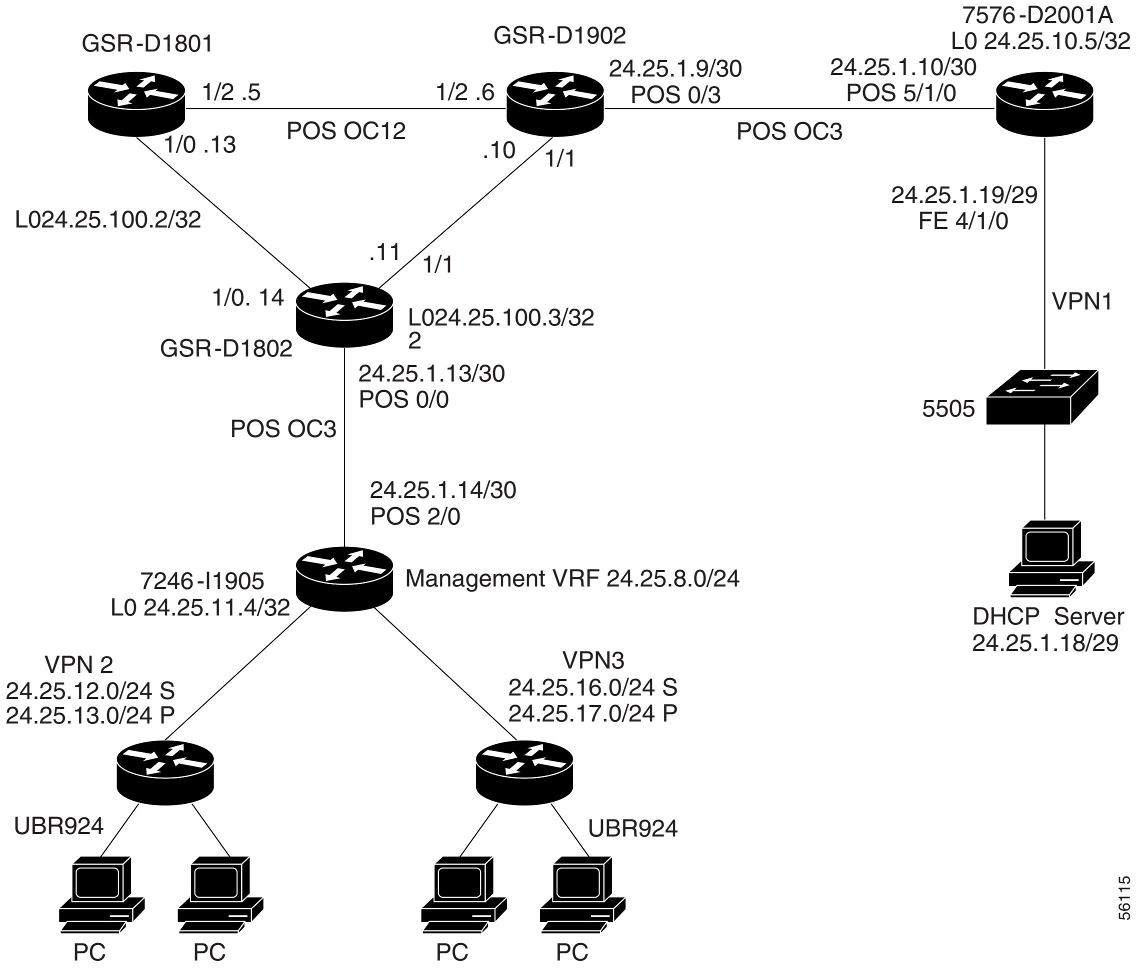
This section describes the following debugging topics:

- Modem and PC Call Flow Sequence, page 4-4
- Troubleshooting Cisco uBR Cable Modems Not Coming Online, page 4-6
- Troubleshooting Cisco uBR7200 VHG/PE Routers, page 4-6
- Troubleshooting the CNR Network Server, page 4-14

The information in this section uses the sample network in Figure 4-2.

Note The troubleshooting steps in this section assume that a management and provisioning VRF has been configured and that the cable modem DHCP server is connected to the management VRF. In the example network in Figure 4-2 and configured as described in the "Troubleshooting Cisco uBR7200 VHG/PE Routers" section on page 4-6, a single Cisco Network Registrar (CNR) DHCP server provides IP addresses to both cable modems and hosts.

Figure 4-2 Sample Cable Access to MPLS VPN Network



Modem and PC Call Flow Sequence

Once MPLS is successfully configured, the cable modems and subscriber PCs should function as described in the following sections:

- Modem Initialization, page 4-4
- Subscriber PC Initialization, page 4-5
- DHCP Renewal, page 4-5
- PC DHCP Renewal Relies on the MPLS-VPN Being in Place, page 4-6

Modem Initialization

When properly configured, the cable modems (CMs) interact with the CMTS subinterfaces as follows:

- The CM initializes and sends a DHCP request.

- The DHCP request is received by the CMTS on the first logical subinterface (for example, Cable 3/0.1).
- The CMTS forwards the DHCP request (based on the cable helper-address settings of interface Cable 3/0.1) to Cisco Network Registrar (CNR).
- CNR uses client-class processing to determine which IP range to give the CM based on the subscriber's Internet Service Provider (ISP) preference.
- CNR sends a DHCP response with all necessary DOCSIS parameters, including an IP address in the range used on the subinterface associated with the subscriber's ISP.
- The CM receives its IP address and other information and begins to send and receive traffic.
- Because the CM has an IP address associated with a specific subinterface, all traffic from that CM is also associated with the subinterface.
- The subinterface is associated with a VRF and, therefore, all traffic to and from the CM travels through the ISP's MPLS-VPN.

Subscriber PC Initialization

When using secondary addresses on each ISP's subinterface, PC IP address acquisition should behave as follows:

- All CM traffic is associated with a specific subinterface. The interface also has a secondary IP address defining a range of ISP IP addresses for subscriber use.
- The subinterface is associated with a VRF and, therefore, all traffic to and from the CM travels through the ISP's MPLS-VPN.
- The subscriber PC sends a DHCP request that is bridged through the CM and sent to the ISP's subinterface.
- The CMTS forwards the DHCP request (based on cable helper-address settings of ISP's subinterface) through the MPLS-VPN to the ISP's network.
- The ISP's DHCP server (CNR) responds to the DHCP request and sends it back through the MPLS-VPN to the appropriate subinterface on the CMTS where it is bridged back to the PC.
- The PC receives its IP address and other DHCP parameters. The PC begins to send and receive traffic.
- Because the subinterface has a secondary IP address range including the PC's IP address, PC traffic is also associated with the appropriate subinterface.
- PC traffic is sent and received via the MPLS-VPN.

DHCP Renewal

Cable modem DHCP renewal requires that the CM have continued access to the Multiservice Operator's (MSO's) CNR server.

- The cable modem lease requires renewal.
- The CM sends a DHCP renew unicast to the MSO's CNR server.
- Because the management VPN is configured to share routes with all other VPNs, the CMTS knows how to route the DHCP renew request directly to the CNR server.

- The CNR server is connected to an interface within the management VRF. Since the management VRF shares its routes with all other VRFs and imports all routes from other VRFs, connectivity between the cable modem and the CNR device exists in both directions.

However, in certain failure situations (for example, partial network outages) the CM may not be able to receive a DHCP renewal. If this is the case, DOCSIS cable modems reinitalize and the complete initialization process begins again with only minor interruption of service.

PC DHCP Renewal Relies on the MPLS-VPN Being in Place

- The PC lease requires renewal
- The PC sends a DHCP renew unicast to the ISP's CNR server.
- The DHCP request is forwarded through the ISP's MPLS-VPN to the ISP's CNR server.
- A DHCP renew response is returned via the MPLS-VPN to the appropriate subinterface and then to the PC.
- If the PC encounters difficulties reaching the ISP's DHCP server, the PC must retry until the server is available.

As soon as a CMTS receives a DHCP response for a cable modem's request, it associates that cable modem with the corresponding subinterface. The DHCP process does not need to fully complete for this to occur. If the DHCP process does not fully complete, the cable modem rebroadcasts a DHCP request. The GIADDR field of the request is set to the IP address of the CMTS subinterface that provides the routing function for the cable modem's assigned IP address, not the IP address of the first logical cable subinterface.

Troubleshooting Cisco uBR Cable Modems Not Coming Online

For information on troubleshooting uBR cable modems, see the tech note at the following URL:

http://www.cisco.com/warp/customer/109/troubleshooting_cm_online.html

This tech note discusses the different states that cable modems (CMs) go through before coming online and establishing IP connectivity. The tech note highlights the most commonly used Cisco IOS troubleshooting commands to verify what state the CM is in and the reasons that can cause the modem to arrive at that state. This is illustrated by **debug** and **show** commands at both the Cable Modem Termination System (CMTS) and the CM. The tech note also discusses some of steps that can be taken to arrive at the correct status, online.

Troubleshooting Cisco uBR7200 VHG/PE Routers

The troubleshooting steps in this section use the sample network in Figure 4-2. Routers 7246-I1904 and 7576-D2001A function as provider edge (PE) routers. The CNR DHCP server is connected to an interface in VPN1. The configurations of the PE routers are described below.

Current configuration:

```
!
version 12.1
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname 7246-I1905
```

```
!

no cable qos permission update
cable qos permission modems
cable time-server
ip subnet-zero
ip cef
no ip finger
no ip domain-lookup
!
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 route-target import 1000:1000
!
ip vrf vpn2
 rd 200:1
 route-target export 1000:1000
 route-target export 200:1
 route-target import 200:1
 route-target import 100:1
!
ip vrf vpn3
 rd 300:1
 route-target export 300:1
 route-target export 1000:1000
 route-target import 100:1
 route-target import 300:1
!
interface Loopback0
 ip address 24.25.11.4 255.255.255.255
!
!

interface POS2/0
 ip address 24.25.1.14 255.255.255.252
 ip route-cache flow
 load-interval 30
 tag-switching ip
 clock source internal
!
interface Cable3/0
 no ip address
 ip route-cache flow
 load-interval 30
 no keepalive
 cable downstream rate-limit token-bucket shaping
 cable downstream annex B
 cable downstream modulation 64qam
 cable downstream interleave-depth 32
 cable downstream frequency 583000000
 cable upstream 0 frequency 37008000
 cable upstream 0 power-level -10
 no cable upstream 0 shutdown
 cable upstream 1 shutdown
 cable upstream 2 shutdown
 cable upstream 3 shutdown
 cable dhcp-giaddr policy
 cable privacy mandatory
 cable privacy kek life-time 3600
 cable privacy tek life-time 3600
!
```

```
interface Cable3/0.1
 description ****** Provisioning and Management *****
 ip vrf forwarding vpn1
 ip address 24.25.8.1 255.255.255.0
 cable dhcp-giaddr policy
 cable helper-address 24.25.1.18
!
interface Cable3/0.2
 description *****VPN2 Cable Modem and Users Subnet****
 ip vrf forwarding vpn2
 ip address 24.25.12.1 255.255.255.0 secondary
 ip address 24.25.13.1 255.255.255.0
 cable dhcp-giaddr policy
 cable helper-address 24.25.1.18
!
interface Cable3/0.3
 description ***** VPN3 Cable Modem and Users Subnet ********
 ip vrf forwarding vpn3
 ip address 24.25.16.1 255.255.255.0 secondary
 ip address 24.25.17.1 255.255.255.0
 cable dhcp-giaddr policy
 cable helper-address 24.25.1.18
!
router ospf 1
 log-adjacency-changes
 network 24.25.1.14 0.0.0.0 area 0

!
router bgp 200
 no synchronization
 bgp log-neighbor-changes
 bgp scan-time 5
 neighbor 24.25.10.4 remote-as 200
 neighbor 24.25.10.4 update-source Loopback0
 neighbor 24.25.10.5 remote-as 200
 neighbor 24.25.10.5 update-source Loopback0
 !
 address-family ipv4 vrf vpn3
 no auto-summary
 no synchronization
 network 24.25.16.0 mask 255.255.255.0
 network 24.25.17.0 mask 255.255.255.0
 exit-address-family
 !
 address-family ipv4 vrf vpn2
 no auto-summary
 no synchronization
 network 24.25.12.0 mask 255.255.255.0
 network 24.25.13.0 mask 255.255.255.0
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 no auto-summary
 no synchronization
 network 24.25.8.0 mask 255.255.255.0
 exit-address-family
 !
 address-family vpnv4
 neighbor 24.25.10.5 activate
 neighbor 24.25.10.5 send-community extended
 bgp scan-time 15
 bgp scan-time import 5
 exit-address-family
!
```

```
ip classless


version 12.1
service timestamps debug datetime msec
service timestamps log uptime
no service password-encryption
service compress-config
no service dhcp
!
hostname 7576-D2001A
!
no logging console

ip subnet-zero
no ip domain-lookup
!
!
!
ip vrf vpn1
 rd 100:1
 route-target export 100:1
 route-target import 100:1
 route-target import 1000:1000
!
ip vrf vpn2
 rd 200:1
 route-target export 1000:1000
 route-target export 200:1
 route-target import 200:1
 route-target import 100:1


ip vrf vpn3
 rd:300:1
 route-target export 300:1
 route-target export 1000:1000
 route-target import 100:1
 route-target import 300:1


ip cef
cns event-service server
!
!
!
interface Loopback0
 ip address 24.25.10.5 255.255.255.255

!
interface FastEthernet4/1/0
description *** Provisioning and Management, CNR Server Attached***
 ip vrf forwarding vpn1
 ip address 24.25.1.17 255.255.255.248
 no ip route-cache distributed
 half-duplex

!
interface POS5/1/0
 ip address 24.25.1.10 255.255.255.252
 no ip route-cache distributed
 tag-switching ip
 crc 32
 clock source internal
 no cdp enable
!
router ospf 1
 log-adjacency-changes
 redistribute static metric 10 subnets
 network 24.25.1.10 0.0.0.0 area 0
 network 24.25.10.5 0.0.0.0 area 0
!
router bgp 200
 no synchronization
 bgp scan-time 5
 neighbor 24.25.10.4 remote-as 200
 neighbor 24.25.10.4 update-source Loopback0
  no auto-summary
 !
 address-family ipv4 vrf vpn2
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf vpn1
 redistribute connected route-map 1_to_world
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family ipv4 vrf forwarding
 no auto-summary
 no synchronization
 exit-address-family
 !
 address-family vpnv4
 neighbor 24.25.10.4 activate
 neighbor 24.25.10.4 send-community extended
 neighbor 24.25.11.4 activate
 neighbor 24.25.11.4 send-community extended
 no auto-summary
 bgp scan-time 5
 bgp scan-time import 5
 exit-address-family
!
ip classless
no ip http server
!
access-list 40 permit 24.25.1.16 0.0.0.7
route-map 1_to_world permit 10
 match ip address 40
!
!

end
```

In this configuration, the first cable subinterface is in a management VPN. Management VPN routes are exported to all other VPNs and the management VPN imports routes from all other VPNs. The CNR DHCP server provides IP addresses for all cable modems and hosts.

Ensure that the users’ VPNs have routes to the DHCP server and that the management VRF on the PE that is connected to the DHCP server has a route to all cable modem and user subnets.

On router 7246-I1905:

```
7246-I1905# show ip route vrf vpn2

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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set:

```
     24.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
B       24.25.8.0/24 is directly connected, 1d18h, Cable3/0.1
C       24.25.13.0/24 is directly connected, Cable3/0.2
C       24.25.12.0/24 is directly connected, Cable3/0.2
B       24.25.1.16/29 [200/0] via 24.25.10.5, 1d19h <<<<<<<<<<<<<<<<
```

On router 7576-D2001A:

```
7576-D2001A# show ip route vrf vpn1

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, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
```

Gateway of last resort is not set:

```
     24.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
B       24.25.3.0/24 [200/0] via 24.25.10.4, 1d19h
B       24.25.2.0/24 [200/0] via 24.25.10.4, 1d19h
B       24.25.4.0/24 [200/0] via 24.25.10.4, 00:00:58
B       24.25.7.0/24 [200/0] via 24.25.10.4, 1d19h
B       24.25.6.0/24 [200/0] via 24.25.10.4, 1d19h
B       24.25.8.0/24 [200/0] via 24.25.11.4, 1d18h
B       24.25.13.0/24 [200/0] via 24.25.11.4, 1d18h<<<<<<<<<<<<<
B       24.25.12.0/24 [200/0] via 24.25.11.4, 1d18h<<<<<<<<<<<<<
B       24.25.17.0/24 [200/0] via 24.25.11.4, 1d18h<<<<<<<<<<<<<
B       24.25.16.0/24 [200/0] via 24.25.11.4, 1d18h<<<<<<<<<<<<
C       24.25.1.16/29 is directly connected, FastEthernet4/1/0
B       24.25.16.0/24 [200/0] via 24.25.11.4, 1d18h
```

If the routing tables are not correct, check that the Border Gateway Protocol (BGP) neighbor relationship is established between the two routers. Ensure that the route export and import rules are correct. If using import or export maps, ensure they are correct.

The **debug ip bgp vpnv4 import** command can be invoked and the neighbor relationship cleared.

Caution Always send **debug** command output to the log file and not to the console. Clearing a BGP neighbor relationship may have detrimental effects on your network. Use with caution and preferably in a lab environment.

```
*Mar 11 08:50:40.803: vpn: 300:1:24.25.6.0 (ver 18), imported as:
*Mar 11 08:50:40.803: vpn: 100:1:24.25.6.0 (ver 22)
*Mar 11 08:50:40.803: vpn: Start import processing for: 300:1:24.25.7.0
*Mar 11 08:50:40.803: vpn: Import check for vpn1; flags mtch
*Mar 11 08:50:40.803: vpn: Import for vpn1 permitted; import flags mtch
*Mar 11 08:50:40.803: BGP: Prefix 300:1:24.25.7.0/24 to be imported as 100:1:24.
25.7.0/24 -- Permitted
nexthop 24.25.10.4, origin i, localpref 100, metric 0, extended community RT:300
:1 RT:1000:1000
```

Once the respective routing tables are correct, use an extended **ping** command to confirm that there is connectivity between the VPN subinterfaces and the relative DHCP servers. In this example, connectivity between router 7246-I1905 VPN2's host subnet and the DHCP server is confirmed.

```
7246-I1905# ping vrf vpn2

Protocol [ip]:
Target IP address: 24.25.1.18
Repeat count [5]: 10
Datagram size [100]: 1500
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 24.25.12.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 10, 1500-byte ICMP Echos to 24.25.1.18, timeout is 2 seconds:
!!!!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 1/3/4 ms
```

If the pings are not successful, check the CEF forwarding table for the remote route:

```
7246-I1905# show ip cef vrf vpn2 24.25.1.18 detail

24.25.1.16/29, version 1286, cached adjacency to POS2/0
0 packets, 0 bytes
  Flow: AS 0, mask 29
  tag information set
    local tag: VPN-route-head
    fast tag rewrite with PO2/0, point2point, tags imposed: {55 45}
  via 24.25.10.5, 0 dependencies, recursive
    next hop 24.25.1.13, POS2/0 via 24.25.10.5/32
    valid cached adjacency
    tag rewrite with PO2/0, point2point, tags imposed: {55 45}
7246-I1905#
```

Note the tags imposed field. The first tag is the tag used to reach the BGP next hop, in this case the loopback interface of router 7576-D2001A. The second tag is the tag advertised to this PE by the remote PE router.

Ensure that the first tag is the correct tag used to reach the BGP next hop. If the outgoing tag is incorrect, the BGP relationship between the two devices should not be established.

```
7246-I1905# show tag forwarding 24.25.10.5 255.255.255.255

Local  Outgoing    Prefix            Bytes tag  Outgoing   Next Hop
tag    tag or VC   or Tunnel Id      switched   interface
63     55          24.25.10.5/32     0          PO2/0      point2point
7246-I1905#
```

Ensure that the BGP table matches the tag displayed by the CEF table. If they are different, record the information and then try clearing the route. Open a case with the Cisco TAC; for more information, see "Obtaining Technical Assistance" in the Preface.

```
7246-I1905# show ip bgp vpnv4 vrf vpn2 tags

   Network          Next Hop      In tag/Out tag
Route Distinguisher: 200:1 (vpn2)
   24.25.1.16/29    24.25.10.5      notag/45
```

On the remote PE router, ensure that the tag it advertises to the local PE matches what the local PE displays. The in tag is the one we are concerned about.

```
7576-D2001A# show ip bgp vpnv4 vrf vpn1 tags

   Network          Next Hop      In tag/Out tag
Route Distinguisher: 100:1 (vpn1)
   24.25.1.16/29    0.0.0.0         45/aggregate(vpn1)
```

If the tags are different, open a case with the Cisco TAC; for more information, see "Obtaining Technical Assistance" in the Preface.

If there is network connectivity between the cable subinterfaces and the DHCP server, the next step is to ensure that the DHCP requests are forwarded to the DHCP servers. In a typical setup, the command **cable dhcp-giaddr policy** is placed on the cable subinterface. This is followed by one or more cable helper addresses. If the keyword modem follows the cable helper command, the DHCP requests from modems are sent to the listed DHCP server. If the keyword host is used, the DHCP requests from the devices attached to the cable modem DHCP requests are forwarded to the listed DHCP server. If neither keyword is used, then all DHCP requests are forwarded to the listed server.

If the **show cable modem** command indicates the cable modem hangs in the init(d) state, this indicates that the cable modem is not getting an answer to its DHCP request.

The initial DHCP request from a modem has its GIADDR field set to the IP address of the first logical cable subinterface. After the initial reply is received from the DHCP server, the CMTS sets the GIADDR to the IP address of the cable subinterface that is to provide routing for the cable modem.

Use the commands **debug ip dhcp server packet**, **debug ip udp**, and **debug ip packet detail** *access-list*, where the *access-list* allows all packets to and from the DHCP server to be recorded. Ensure that **debug** command output is sent to the log only by executing the **no logging console** command.

The DHCP process involves several steps. The host broadcasts a DHCP discover packet. The router forwards this request to the appropriate DHCP servers. The DHCP server returns a DHCP offer to the IP address contained in the GIADDR field of the DHCP discover packet. The router forwards the offer to the MAC address of the requesting station. The host then broadcasts a DHCP request asking that it be awarded the IP address that was in the DHCP offer. The router forwards this to the DHCP servers. If the host received multiple DHCP offers, the other servers know if their offers were accepted or not. The DHCP server whose offer was accepted sends a DHCP acknowledgement to confirm that the IP address offered is valid for the duration of the lease.

The **debug** commands below show that the initial DHCP discover packet GIADDR field is set to the IP address of the first cable subinterface (24.25.8.1). After the router forwards the initial offer, the GIADDR field of subsequent packets is set to the IP address of the router interface that provides routing for the cable modem.

```
7246-I1905# show access-list 150

Extended IP access list 150
    permit udp any host 24.25.1.18
    permit udp host 24.25.1.18 any

7246-I1905# debug ip packet detail 150

IP packet debugging is on (detailed) for access list 150

7246-I1905# debug ip udp

UDP packet debugging is on

7246-I1905# debug ip dhcp server packet
```

```
00:06:09: BOOTP: opcode 1 on interface Cable3/0.1, 0 secs, 0 hops
00:06:09: DHCPD: setting giaddr to 24.25.8.1.
00:06:09: UDP: sent src=24.25.8.1(67), dst=24.25.1.18(67), length=604
00:06:09: datagramsize=576, IP 0: s=24.25.8.1 (local), d=24.25.1.18, totlen 604,
 fragment 0, fo 0, cef process switched
00:06:09:     UDP src=67, dst=67
00:06:09: datagramsize=576, IP 0: s=24.25.8.1 (local), d=24.25.1.18, totlen 604,
 fragment 0, fo 0, cef process switched
00:06:09:     UDP src=67, dst=67
00:06:09: DHCPD: BOOTREQUEST from 0100.3080.ea8e.53 forwarded to 24.25.1.18.
00:06:09: UDP: rcvd src=24.25.1.18(67), dst=24.25.8.1(67), length=308
00:06:09: DHCPD: forwarding BOOTREPLY to client 0030.80ea.8e53.
00:06:09: DHCPD: broadcasting BOOTREPLY to client 0030.80ea.8e53.
00:06:09: UDP: sent src=0.0.0.0(67), dst=255.255.255.255(68), length=328
00:06:09: UDP: rcvd src=0.0.0.0(68), dst=255.255.255.255(67), length=584
00:06:09: BOOTP: opcode 1 on interface Cable3/0.3, 0 secs, 0 hops
00:06:09: DHCPD: setting giaddr to 24.25.17.1.
00:06:09: UDP: sent src=24.25.17.1(67), dst=24.25.1.18(67), length=604
00:06:09: datagramsize=576, IP 2: s=24.25.17.1 (local), d=24.25.1.18, totlen 604
, fragment 0, fo 0, cef process switched
00:06:09:     UDP src=67, dst=67
00:06:09: datagramsize=576, IP 2: s=24.25.17.1 (local), d=24.25.1.18, totlen 604
, fragment 0, fo 0, cef process switched
00:06:10:     UDP src=67, dst=67
00:06:10: UDP: rcvd src=24.25.1.18(67), dst=24.25.17.1(67), length=308
00:06:10: DHCPD: forwarding BOOTREPLY to client 0030.80ea.8e53.
00:06:10: DHCPD: broadcasting BOOTREPLY to client 0030.80ea.8e53.
00:06:10: UDP: sent src=0.0.0.0(67), dst=255.255.255.255(68), length=328
```

Ensure that the DHCP request is forwarded to the correct server IP address and that an answer is forwarded to the cable modem. If no answer is received or the answer is received and forwarded but the cable modem still stays in the init(d) state, troubleshoot the CNR DHCP server.

Troubleshooting the CNR Network Server

Typically the CNR server is configured to use client class processing for cable modem DHCP requests. The client's MAC address is placed into a client class that has a scope selection tag defined. The selection tag matches those assigned to the desired scope. Since the initial DHCP request for all cable modems arrives with the GIADDR field set to the IP address of the first logical cable subinterface, secondary scopes must be used. A typical configuration is to have a scope whose IP address range includes the first logical cable subinterface configured as a primary scope. The actual IP addresses to be assigned to the cable modems are contained within scopes that are secondary to the first scope. The correct secondary scope is selected because it has scope selection tags that match the ones defined by the client class.

When the initial DHCP discover arrives, the primary scope is selected based on the GIADDR. The primary scope can be configured with only a single IP address and this IP address could be reserved to a bogus MAC address. The CNR server then uses client class processing to select the desired scope that is a secondary to this primary.

To eliminate the need to configure each and every MAC address into a client class, you can define a default client class. The MAC address contained within the client class would be the default. This MAC address matches all MAC addresses that are not defined in another client class. Attach a scope selection tag to this client class and to all scopes that can provide IP addresses to clients (both cable modem and end user hosts) whose MAC addresses are not listed in client classes.

The first step in troubleshooting CNR problems is to ensure that the DHCP request is getting to the CNR server and that the GIADDR field of the request is as expected.

```
$ cd /opt/nwreg2/usrbin
$ ./nrcmd -s
100 Ok
session:
    cluster = localhost
    default-format = user
    user-name = admin
    visibility = 5

nrcmd> server dhcp setDebug 10

100 Ok

nrcmd>


03/11/2001 13:39:57 name/dhcp/1 Activity Server 0 04619 Server received a DHCPDI
SCOVER packet 'R49173' from: CID: 01:00:30:80:ea:8e:53  via: Gateway 24.25.8.1,
1 in use.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'merit-dump', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'host-name', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'boot-file', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'log-servers', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'domain-name', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'domain-name-servers', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Info Configuration 0 04630 The server could not
return option 'mcns-security-server', which a client requested.
03/11/2001 13:39:57 name/dhcp/1 Activity Protocol 0 04993 24.25.17.101 Lease off
ered to CID: 01:00:30:80:ea:8e:53  packet 'R49173'  until Sun, 11 Mar 2001 13:41
:57 -0500. 4 ms.
```

INDEX