CHAPTER 17

Troubleshooting ISDN Connections

Introduction

Dialup is simply the application of the Public Switched Telephone Network (PSTN) to carry data on behalf of the end user. It involves customer premises equipment (CPE) sending the telephone switch a phone number to direct a connection to. The Cisco 3600, AS5200, AS5300, and AS5800 are all examples of routers that have the capability to run a PRI along with banks of digital modems. The AS2511, on the other hand, is an example of a router that communicates with external modems.

Since the time of the last *Internetworking Troubleshooting Handbook*, the carrier market has grown significantly, and there has been a demand for higher modem densities. The answer to this need was a higher degree of interoperation with the telco equipment and the development of the digital modem. This type of modem is capable of direct digital access to the PSTN. This has allowed the development of faster CPE modems that can take advantage of the clarity of signal that the digital modems enjoy. The fact that digital modems connecting into the PSTN through a PRI or BRI can transmit data at greater than 53K using the V.90 communication standard attests to the success of the idea.

The first Cisco access servers were the Cisco 2509 and Cisco 2511. The 2509 could support 8 incoming connections using external modems, and the 2511 could support 16. The AS5200 was introduced with 2 PRIs and could support 48 users using digital modems, which represented a major leap forward in technology. Modem densities have increased steadily, with the AS5300 supporting four and then eight PRIs. Recently, the AS5800 was introduced to fill the needs of carrier class installations that needed to handle dozens of incoming T1s and hundreds of user connections.

A few outdated technologies bear mentioning in a historical discussion of dialer technology. 56K flex is an older (pre-V.90) 56K modem standard that was proposed by Rockwell. Cisco supports version 1.1 of the 56K flex standard on its internal modems, but the company recommends migrating the CPE modems to V.90 as soon as possible. Another outdated technology is the AS5100, which was a joint venture between Cisco and a modem manufacturer. Created as a way to increase modem density through the use of quad modem cards, the AS5100 involved a group of 2511s built as cards that inserted into a backplane shared by quad modem cards and a dual T1 card.

Troubleshooting Incoming Calls

Troubleshooting an incoming call starts at the bottom—the physical layer—and works up the protocol stack. The general flow of reasoning looks for the following (a “yes” answer advances to the next question):

- Do we see the call arrive?
- Does the receiving end answer the call?

- Does the call complete?
- Is data passing across the link?
- Is the session established? (PPP or terminal)

For modem connections, a data call looks the same as a terminal session coming in until the end, when the data call goes to negotiate PPP.

For incoming calls involving digital modems, first make sure that the underlying ISDN or CAS is receiving the call. If you are using an external modem, you can skip the ISDN and CAS group sections.

Incoming ISDN Call Troubleshooting

Use the command **debug isdn q931** to watch the q931 signaling messages go back and forth while the router negotiates the ISDN connection. Here's an example output from a successful connection:

```
Router# debug isdn q931
RX <- SETUP pd = 8 callref = 0x06
 Bearer Capability i = 0x8890
 Channel ID i = 0x89
 Calling Party Number i = 0x0083, `5551234'
TX -> CONNECT pd = 8 callref = 0x86
RX <- CONNECT_ACK pd = 8 callref = 0x06
```

The SETUP message indicates that a connection is being initiated by the remote end. The call reference numbers are maintained as a pair. In this case, the call reference number for the incoming side of the connection is 0x06, while the call reference number of the outbound side of the connection is 0x86. The bearer capability (often referred to as the bearercap) tells the router what kind of call is coming in. In this case, the connection is type 0x8890. That value indicates "ISDN speed 64 kbps." If the bearercap had been 0x8090A2, it would have indicated "Speech/voice call u-law."

If no setup message was seen coming in, verify the correct number (try calling it manually, if it is voice-provisioned) and check the status of the ISDN interface (see Chapter 16, "Troubleshooting Dialup Connections"*).* If all that checks out, make sure that the call originator is making the correct call. Contact the telco to trace the call to see where it's being sent. If the connection is a long-distance one, try a different long-distance carrier using a 1010 long-distance code.

If the call coming in is an async modem call, make sure that the line is provisioned to allow voice calls.

Note BRI async modem calling is a feature of 3600 routers running 12.0(3)T or later. It requires a recent hardware revision of the BRI interface network module. WIC modules do not support async modem calling.

If the call arrived but did not complete, look for a cause code (see Table 17-10). A successful completion is shown by a connect-ack being issued.

If this is an async modem call, move forward to the "Incoming Modem Call Troubleshooting" section.

At this point, the ISDN call is connected, but no data has been seen coming across the link. Use the command **debug ppp negotiate** to see whether any PPP traffic is coming across the line. If not, there may be a speed mismatch. To determine whether this is the case, use the **show running-config** privileged exec command to view the router configuration. Check the **dialer map** interface configuration command entries in the local and remote router. These entries should look similar to the following:

dialer map ip 131.108.2.5 speed 56 name C4000

For dialer profiles, a map class must be defined to set the speed. Note that, by default, ISDN interfaces attempt to use 64K communications speeds on each channel.

For detailed information on configuring dialer maps and profiles, refer to the *Cisco IOS Dial Solutions Configuration Guide*, *Dial Solutions Command Reference*, and *Dial Solutions Quick Configuration Guide.*

Getting valid PPP packets indicates that the link is up and working. Proceed to the section "Troubleshooting PPP."

Incoming CAS Call Troubleshooting

To troubleshoot the CAS group serving connectivity to the modems, use the commands **debug modem**, **debug modem csm**, and **debug cas**.

Note The **debug cas** command first appeared in 12.0(7)T for the AS5200 and AS5300. Earlier versions of IOS use a system-level configuration command **service internal**, along with the exec command **modem-mgmt debug rbs**. Debugging this information on an AS5800 requires connecting to the trunk card itself.

The first thing to look for is if the telco switch went off-hook to signal the incoming call. If not, verify the number being called by attaching a phone to the originating side's phone line and then calling the number. If the call comes in, the problem is in the originating CPE. If the call still does not show up on the CAS, check the T1 (per Chapter 15, "Troubleshooting Serial Lines"). A good debug to use in this instance is **debug serial interfaces.**

The following shows a good connection using **debug modem CSM**:

```
Router# debug modem csm
CSM_MODEM_ALLOCATE: slot 1 and port 0 is allocated.
MODEM_REPORT(0001): DEV_INCALL at slot 1 and port 0
CSM_PROC_IDLE: CSM_EVENT_ISDN_CALL at slot 1, port 0
CSM_RING_INDICATION_PROC: RI is on
CSM_RING_INDICATION_PROC: RI is off
CSM_PROC_IC1_RING: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
MODEM_REPORT(0001): DEV_CONNECTED at slot 1 and port 0
CSM_PROC_IC2_WAIT_FOR_CARRIER: CSM_EVENT_ISDN_CONNECTED at slot 1, port 0
```

If this code is seen, then the call was directed to a modem. Proceed to the "Incoming Modem Call Troubleshooting" section.

Incoming Modem Call Troubleshooting

The debugs used in troubleshooting incoming modem calls are listed here:

debug modem

debug modem csm (for integrated digital modems)

These other debugs are used in conjunction to indicate the new call coming in:

debug isdn q931

debug cas

Assuming that the call reaches the modem, the modem must pick up the call.

Tips for Debugging External Modems

To facilitate debugging on an external modem connected to a TTY line, turn up the speaker volume—it helps make some problems more apparent.

When the originating modem calls, does the receiving modem ring? If not, verify the number and try a manual call from the remote site. Try using a regular phone on the receiving end as well. Replace cables and hardware as needed.

Async Modem Call Pickup

If an external modem is not answering, check the cabling between the modem and the access server or router. Confirm that the modem is connected to the TTY or auxiliary port on the router with a rolled RJ-45 cable and an MMOD DB-25 adapter. This cabling configuration is recommended and supported by Cisco for RJ-45 ports. (These connectors are typically labeled “Modem.”)

The most common types of RJ-45 cabling are straight, rolled, and crossover. If you hold the two ends of an RJ-45 cable side by side, you’ll see eight colored strips, or pins, at each end. If the order of the colored pins is the same at each end, then the cable is straight. If the order of the colors is reversed at each end, then the cable is rolled. The cable is a crossover cable if colors indicate the following:

```
RJ45 to RJ45 crossover cable
      RJ45                    RJ45
         5 ------------------ 2

         2 ------------------ 5

         4 ------------------ 1

         1 ------------------ 4
```

To make sure that the signaling is okay, use the **show line** command outlined in Chapter 16.

Cabling issues aside, an external modem must be initialized to autoanswer. Check the remote modem to see whether it is set to autoanswer. Usually, an AA indicator light is on when autoanswer is set. Set the remote modem to autoanswer if it is not already set. To find out how to verify and change the modem’s settings, refer to your modem documentation. Use a reverse telnet (See Chapter 16) to initialize the modem.

Digital (Integrated) Modem Call Pickup

On an external modem, it is clear whether the call is getting answered, but internal modems require a manual call to the receiving number. Listen for the answer back tone (ABT). If no ABT is heard, check the configuration for two things:

- Make sure that the command **isdn incoming-voice modem** exists under any ISDN interfaces handling incoming modem connections.
- Under the line configuration for the modem’s TTY, make sure that the command **modem inout** exists.

It is also possible that an internal modem was not allocated by the Call Switching Module (CSM) to handle the incoming call. This problem can be caused by modem or resource pools being configured for too few incoming connections, or the access server my simply be out of modems. Check the availability of modems, and adjust the modem pool or resource pool manager settings appropriately. If a modem was allocated and the configuration shows **modem inout,** gather debugs and contact Cisco for assistance.

Modem Trainup

A successful trainup is indicated by the receiving modem raising DSR. Trainup failures can indicate a circuit problem or modem incompatibility.

If you really want to get to the bottom of an individual modem problem, you'll want to get your hands to the AT prompt at the originating modem, while it's attached to the POTS line of interest. If you're calling into a digital modem in a Cisco access server, be prepared to record a .wav file of the trainup "music," or Digital Impairment Learning (DIL) sequence. The DIL is the musical score (PCM sequence) that the originating V.90 analog modem tells the receiving digital modem to play back. The sequence allows the analog modem to discern any digital impairment in the circuit (such as multiple D/A conversions, a-law/u-law, robbed bits, and digital pads). If you don't hear the DIL, the modems did not negotiate V.90 in V.8/V.8bis (that is, a modem compatibility issue has arisen). If you *do* hear the DIL, but then you hear a retrain in V.34, the analog modem has decided, on the basis of the DIL playback, that V.90 was infeasible.

Does the music have noise in it? If so, then clean up the circuit.

Does the client give up quickly, without running V.34 training? Perhaps it doesn't know what to do when it hears V.8bis. Try disabling V.8bis (hence, 56K flex) on the server (if acceptable), getting new client firmware, or swapping out the client modem. Alternately, the dialing end could insert five commas at the end of the dial string. This delays the calling modem's listening and causes the V.8bis tone from the receiving server to time out without affecting the client modem. Five commas in the dial string is a ballpark estimate, though, so you might need to adjust this to allow for local conditions.

Session Establishment

At this point in the sequence, the modems are connected and trained up. Now it's time to find out whether any traffic is coming across properly.

If the line receiving the call is configured with **autoselect ppp** and the async interface is configured with **async mode interactive**, use the command **debug modem** to verify the autoselect process. As traffic comes in over the async link, the access server examines the traffic to determine whether the traffic is character-based or packet-based. Depending on the determination, the access server then either starts a PPP session or goes no farther than having an exec session on the line.

This is a normal autoselect sequence with inbound PPP LCP packets:

```
*Mar  1 21:34:56.958: TTY1: DSR came up
*Mar  1 21:34:56.962: tty1: Modem: IDLE->READY
*Mar  1 21:34:56.970: TTY1: EXEC creation
*Mar  1 21:34:56.978: TTY1: set timer type 10, 30 seconds
*Mar  1 21:34:59.722: TTY1: Autoselect(2) sample 7E            Note 1
*Mar  1 21:34:59.726: TTY1: Autoselect(2) sample 7EFF
*Mar  1 21:34:59.730: TTY1: Autoselect(2) sample 7EFF7D
*Mar  1 21:34:59.730: TTY1: Autoselect(2) sample 7EFF7D23
*Mar  1 21:34:59.734: TTY1 Autoselect cmd: ppp negotiate       Note 2
*Mar  1 21:34:59.746: TTY1: EXEC creation
*Mar  1 21:34:59.746: TTY1: create timer type 1, 600 seconds
*Mar  1 21:34:59.794: TTY1: destroy timer type 1 (OK)
*Mar  1 21:34:59.794: TTY1: destroy timer type 0
*Mar  1 21:35:01.798: %LINK-3-UPDOWN: Interface Async1, changed state to up    Note 3
```

Note 1: The inbound traffic is displayed in hexadecimal format, based on the bits coming in over the line, regardless of whether the bits are ASCII characters or elements of a packet. The bits represented in this example are correct for an LCP packet. Anything different would be either a malformed packet or character traffic.

Note 2: Having determined that the inbound traffic is actually an LCP packet, the access server triggers the PPP negotiation process.

Note 3: The async interface changes state to up, and the PPP negotiation (not shown) commences.

If the call is a PPP session, and if **async mode dedicated** is configured on the async interface, use the command **debug ppp negotiation** to see whether any configuration request packets (the debugs will show them as CONFREQ) are coming from the remote end. If PPP packets are seen to be both inbound and outbound, proceed to the section "PPP Debugging." Otherwise, connect in from the call-originating end with a character-mode (or "exec") session (that is, a non-PPP session).

Note If the receiving end has the setting **async modem dedicated** under the async interface, an exec dial-in will see nothing but what appears to be random ASCII garbage. To allow a terminal session and still have PPP capability, use the async interface configuration command **async mode interactive**. Under the associated line's configuration, use the command **autoselect ppp**.

If the modems connect with a terminal session and no data comes across, check the causes shown in Table 17-1.

Table 17-1 Modem Cannot Send or Receive Data

Possible Causes	Suggested Actions
Modem speed setting is not locked	1. Use the **show line** exec command on the access server or router. The output for the auxiliary port should indicate the currently configured Tx and Rx speeds. 2. If the line is not configured to the correct speed, use the **speed** line configuration command to set the line speed on the access server or router line. Set the value to the highest speed in common between the modem and the access server or router port.

Table 17-1 Modem Cannot Send or Receive Data (continued)

Possible Causes	Suggested Actions
Modem speed setting is not locked *(continued)*	To set the terminal baud rate, use the **speed** line configuration command. This command sets both the transmit (to terminal) and receive (from terminal) speeds. Syntax: **speed** *bps* Syntax Description: • *bps*—Baud rate in bits per second (bps). The default is 9600 bps. Example: The following example sets lines 1 and 2 on a Cisco 2509 access server to 115200 bps: **line 1 2** **speed 115200** **Note:** If you cannot use flow control for some reason, limit the line speed to 9600 bps. Faster speeds likely will result in lost data. 3. Use the **show line** exec command again, and confirm that the line speed is set to the desired value. 4. When you are certain that the access server or router line is configured for the desired speed, initiate a reverse Telnet session to the modem via that line. For more information, see the section "Establishing a Reverse Telnet Session to a Modem," in Chapter 16. 5. Use a modem command string that includes the lock DTE speed command for your modem. See your modem documentation for exact configuration command syntax. **Note:** The modem AT command to lock DTE speed, which might also be referred to as *port rate adjust* or *buffered mode*, is often related to the way in which the modem handles error correction. This command varies widely from one modem to another. Locking the modem speed ensures that the modem always communicates with the Cisco access server or router at the speed configured on the Cisco auxiliary port. If this command is not used, the modem reverts to the speed of the data link (the telephone line) instead of communicating at the speed configured on the access server.

Table 17-1 Modem Cannot Send or Receive Data (continued)

Possible Causes	Suggested Actions
Hardware flow control is not configured on the local or remote modem or router	1. Use the **show line** *aux-line-number* exec command, and look for the following in the Capabilities field (see the section "Interpreting Show Line Output" in Chapter 16): **Capabilities: Hardware Flowcontrol In, Hardware Flowcontrol Out** If there is no mention of hardware flow control in this field, hardware flow control is not enabled on the line. Hardware flow control for access server-to-modem connections is recommended. For an explanation of the output of the **show line** command, see the section "Using **debug** Commands," earlier in this chapter. 2. Configure hardware flow control on the line using the **flowcontrol hardware** line configuration command. To set the method of data flow control between the terminal or other serial device and the router, use the **flowcontrol** line configuration command. Use the **no** form of this command to disable flow control. Syntax: **flowcontrol** {**none** | **software** [**lock**] [**in** | **out**] | **hardware** [**in** | **out**]} Syntax description: • **none**—Turns off flow control. • **software**—Sets software flow control. An optional keyword specifies the direction: **in** causes the Cisco IOS software to listen to flow control from the attached device, and **out** causes the software to send flow control information to the attached device. If you do not specify a direction, both are assumed. • **lock**—Makes it impossible to turn off flow control from the remote host when the connected device needs software flow control. This option applies to connections using the Telnet or rlogin protocols. • **hardware**—Sets hardware flow control. An optional keyword specifies the direction: **in** causes the software to listen to flow control from the attached device, and **out** causes the software to send flow control information to the attached device. If you do not specify a direction, both are assumed. For more information about hardware flow control, see the hardware manual that was shipped with your router.

Table 17-1 Modem Cannot Send or Receive Data (continued)

Possible Causes	Suggested Actions
Hardware flow control is not configured on the local or remote modem or router *(continued)*	Example: The following example sets hardware flow control on line 7: **line 7** **flowcontrol hardware** **Note:** If you cannot use flow control for some reason, limit the line speed to 9600 bps. Faster speeds likely will result in lost data. 3. After enabling hardware flow control on the access server or router line, initiate a reverse Telnet session to the modem via that line. For more information, see the section "Establishing a Reverse Telnet Session to a Modem." 4. Use a modem command string that includes the **RTS/CTS Flow** command for your modem. This command ensures that the modem is using the same method of flow control (that is, hardware flow control) as the Cisco access server or router. See your modem documentation for exact configuration command syntax.
dialer map commands are misconfigured	1. Use the **show running-config** privileged exec command to view the router configuration. Check the **dialer map** command entries to see whether the **broadcast** keyword is specified. 2. If the keyword is missing, add it to the configuration. Syntax: **dialer map** *protocol next-hop-address* [**name** *hostname*] [**broadcast**] [*dial-string*] Syntax description: • *protocol*—The protocol subject to mapping. Options include IP, IPX[1], bridge, and snapshot. • *next-hop-address*—The protocol address of the opposite site's async interface. • **name** *hostname*—A required parameter used in PPP authentication. It is the name of the remote site for which the dialer map is created. The name is case-sensitive and must match the host name of the remote router. • **broadcast**—An optional keyword that broadcast packets (such as IP RIP or IPX RIP/SAP updates) to be forwarded to the remote destination. In static routing sample configurations, routing updates are not desired and the **broadcast** keyword is omitted.

Table 17-1 Modem Cannot Send or Receive Data (continued)

Possible Causes	Suggested Actions
dialer map commands are misconfigured *(continued)*	• *dial-string*—The remote site's phone number. Any access codes (for example, 9 to get out of an office; international dialing codes; and area codes) must be included. 3. Make sure that **dialer map** commands specify the correct next-hop addresses. 4. If the next-hop address is incorrect, change it using the **dialer map** command. 5. Make sure that all other options in **dialer map** commands are correctly specified for the protocol that you are using. For detailed information on configuring dialer maps, refer to the Cisco IOS *Wide-Area Networking Configuration Guide* and *Wide-Area Networking Command Reference*.
A problem has occurred with the dialing modem	Make sure that the dialing modem is operational and is securely connected to the correct port. See whether another modem works when connected to the same port.

Debugging an incoming exec session generally falls into a few main categories. Possible causes and suggested actions can be found in Tables 17-2 through 17-5.

- Dialup client receives no exec prompt
- Dialup session sees "garbage"
- Dialup session ends up in an existing session
- Dialup receiving modem does not disconnect properly

Table 17-2 Dialup Client Receives No exec Prompt

Possible Causes	Suggested Actions
Autoselect is enabled on the line	Attempt to access exec mode by issuing a carriage return.
Line is configured with the **no exec** command	1. Use the **show line** exec command to view the status of the appropriate line. Check the Capabilities field to see whether it include "exec suppressed." If this is the case, the **no exec** line configuration command is enabled. 2. Configure the **exec** line configuration command on the line to allow exec sessions to be initiated. This command has no arguments or keywords. Example: The following example turns on the exec on line 7: **line 7** **exec**

Table 17-2 Dialup Client Receives No exec Prompt (continued)

Possible Causes	Suggested Actions
Flow control is not enabled, is enabled only on one device (either DTE or DCE), or is misconfigured	1. Use the **show line** *aux-line-number* exec command, and look for the following in the Capabilities field (see the section "Interpreting Show Line Output" in Chapter 16): **Capabilities: Hardware Flowcontrol In, Hardware Flowcontrol Out** If there is no mention of hardware flow control in this field, hardware flow control is not enabled on the line. Hardware flow control for access server-to-modem connections is recommended. 2. Configure hardware flow control on the line using the **flowcontrol hardware** line configuration command. Example: The following example sets hardware flow control on line 7: **line 7** **flowcontrol hardware** **Note** If you cannot use flow control for some reason, limit the line speed to 9600 bps. Faster speeds likely will result in lost data. 3. After enabling hardware flow control on the access server or router line, initiate a reverse Telnet session to the modem via that line. For more information, see the section "Establishing a Reverse Telnet Session to a Modem," earlier in this chapter. 4. Use a modem command string that includes the **RTS/CTS Flow** command for your modem. This command ensures that the modem is using the same method of flow control (that is, hardware flow control) as the Cisco access server or router. See your modem documentation for exact configuration command syntax. Figure 16-1 shows the hardware flow control command string for a Hayes-compatible modem.

Table 17-2 Dialup Client Receives No exec Prompt (continued)

Possible Causes	Suggested Actions
Modem speed setting is not locked	1. Use the **show line** exec command on the access server or router. The output for the auxiliary port should indicate the currently configured Tx and Rx speeds. For an explanation of the output of the **show** line command, see the section "Using **debug** Commands," earlier in this chapter. 2. If the line is not configured to the correct speed, use the **speed** line configuration command to set the line speed on the access server or router line. Set the value to the highest speed in common between the modem and the access server or router port.

Table 17-2 Dialup Client Receives No exec Prompt (continued)

Possible Causes	Suggested Actions
Modem speed setting is not locked *(continued)*	To set the terminal baud rate, use the **speed** line configuration command. This command sets both the transmit (to terminal) and receive (from terminal) speeds. Syntax: **speed** *bps* Syntax description: • *bps*—Baud rate in bits per second (bps). The default is 9600 bps. Example: The following example sets lines 1 and 2 on a Cisco 2509 access server to 115200 bps: **line 1 2** **speed 115200** **Note** If you cannot use flow control for some reason, limit the line speed to 9600 bps. Faster speeds likely will result in lost data. 3. Use the **show line** exec command again, and confirm that the line speed is set to the desired value. 4. When you are certain that the access server or router line is configured for the desired speed, initiate a reverse Telnet session to the modem via that line. For more information, see the section "Establishing a Reverse Telnet Session to a Modem," earlier in this chapter. 5. Use a modem command string that includes the lock DTE speed command for your modem. See your modem documentation for exact configuration command syntax. **Note:** The lock DTE speed command, which might also be referred to as port rate adjust or buffered mode, is often related to the way in which the modem handles error correction. This command varies widely from one modem to another. Locking the modem speed ensures that the modem always communicates with the Cisco access server or router at the speed configured on the Cisco auxiliary port. If this command is not used, the modem reverts to the speed of the data link (the telephone line) instead of communicating at the speed configured on the access server.

Table 17-3 Dialup Sessions Sees "Garbage"

Possible Causes	Suggested Actions
Modem speed setting is not locked	1. Use the **show line** exec command on the access server or router. The output for the auxiliary port should indicate the currently configured Tx and Rx speeds. 2. If the line is not configured to the correct speed, use the **speed** line configuration command to set the line speed on the access server or router line. Set the value to the highest speed in common between the modem and the access server or router port. To set the terminal baud rate, use the **speed** line configuration command. This command sets both the transmit (to terminal) and receive (from terminal) speeds. Syntax: **speed** *bps* Syntax description: • *bps*—Baud rate in bits per second (bps). The default is 9600 bps. Example: The following example sets lines 1 and 2 on a Cisco 2509 access server to 115200 bps: **line 1 2** **speed 115200** **Note** If you cannot use flow control for some reason, limit the line speed to 9600 bps. Faster speeds likely will result in lost data. 3. Use the **show line** exec command again, and confirm that the line speed is set to the desired value. 4. When you are certain that the access server or router line is configured for the desired speed, initiate a reverse Telnet session to the modem via that line. For more information, see the section "Establishing a Reverse Telnet Session to a Modem," earlier in this chapter. 5. Use a modem command string that includes the lock DTE speed command for your modem. See your modem documentation for exact configuration command syntax. **Note:** The **lock** DTE speed command, which might also be referred to as *port rate adjust* or *buffered mode*, is often related to the way in which the modem handles error correction. This command varies widely from one modem to another.

Table 17-3 Dialup Sessions Sees "Garbage" (continued)

Possible Causes	Suggested Actions
Modem speed setting is not locked *(continued)*	Locking the modem speed ensures that the modem always communicates with the Cisco access server or router at the speed configured on the Cisco auxiliary port. If this command is not used, the modem reverts to the speed of the data link (the telephone line) instead of communicating at the speed configured on the access server.

Symptom: Remote dial-in session ends up in an already existing session initiated by another user. That is, instead of getting a login prompt, a dial-in user sees a session established by another user (which might be a UNIX command prompt, a text editor session, and so forth).

Table 17-4 Dialup Session Ends Up in Existing Session

Possible Causes	Suggested Actions
Modem configured for DCD is always high	1. The modem should be reconfigured to have DCD high only on CD. This is usually accomplished by using the **&C1** modem command string, but check your modem documentation for the exact syntax for your modem. 2. You might have to configure the access server line to which the modem is connected with the **no exec** line configuration command. Clear the line with the **clear line** privileged exec command, initiate a reverse Telnet session with the modem, and reconfigure the modem so that DCD is high only on CD. 3. End the Telnet session by entering **disconnect**, and reconfigure the access server line with the **exec** line configuration command.

Table 17-4 Dialup Session Ends Up in Existing Session (continued)

Possible Causes	Suggested Actions
Modem control is not enabled on the access server or router	1. Use the **show line** exec command on the access server or router. The output for the auxiliary port should **show inout** or **RIisCD** in the Modem column. This indicates that modem control is enabled on the line of the access server or router. 2. Configure the line for modem control using the **modem inout** line configuration command. Modem control is now enabled on the access server. **Note:** Be certain to use the **modem inout** command instead of the **modem ri-is-cd** command while the connectivity of the modem is in question. The latter command allows the line to accept incoming calls only. Outgoing calls will be refused, making it impossible to establish a Telnet session with the modem to configure it. If you want to enable the **modem ri-is-cd** command, do so only after you are certain that the modem is functioning correctly.
Cabling is incorrect	1. Check the cabling between the modem and the access server or router. Confirm that the modem is connected to the auxiliary port on the access server or router with a rolled RJ-45 cable and an MMOD DB-25 adapter. This cabling configuration is recommended and supported by Cisco for RJ-45 ports. (These connectors are typically labeled "Modem.") Two types of RJ-45 cabling are commonly encountered when connecting a modem: straight and rolled. If you hold the two ends of an RJ-45 cable side by side, you'll see eight colored strips, or pins, at each end. If the order of the colored pins is the same at each end, then the cable is straight. If the order of the colors is reversed at each end, then the cable is rolled. The rolled cable (CAB-500RJ) is standard with Cisco's 2500/CS500. 2. Use the **show line** exec command to verify that the cabling is correct.

continues

Table 17-5 Dialup Receiving Modem Does Not Disconnect Properly

Possible Causes	Suggested Actions
Modem is not sensing DTR	Enter the **Hangup DTR** modem command string. This command tells the modem to drop the carrier when the DTR signal is no longer being received. On a Hayes-compatible modem, the **&D3** string is commonly used to configure **Hangup DTR** on the modem. For the exact syntax of this command, see the documentation for your modem.
Modem control is not enabled on the router or access server	1. Use the **show line** exec command on the access server or router. The output for the auxiliary port should show **inout** or **RIisCD** in the Modem column. This indicates that modem control is enabled on the line of the access server or router. 2. Configure the line for modem control using the **modem inout** line configuration command. Modem control is now enabled on the access server.
Modem control is not enabled on the router or access server *(continued)*	**Note:** Be certain to use the **modem inout** command instead of the **modem dialin** command while the connectivity of the modem is in question. The latter command allows the line to accept incoming calls only. Outgoing calls will be refused, making it impossible to establish a Telnet session with the modem to configure it. If you want to enable the **modem dialin** command, do so only after you are certain that the modem is functioning correctly.

Troubleshooting Outbound Calls

The troubleshooting approach for incoming calls starts at the bottom, but troubleshooting an outbound connection starts at the top. Outbound connection troubleshooting goes along these lines (a "yes" answer to the question gets to the next question):

- Does dial-on-demand routing initiate a call?
- If this is an async modem, do the chat scripts issue the expected commands?
- Does the call make it out to the PSTN?
- Does the remote end answer the call?
- Does the call complete?
- Is data passing over the link?
- Is the session established? (PPP or terminal)

Verifying Dialer Operation

To see whether the dialer is trying to make a call to its remote destination, use the command **debug dialer events**. More detailed information can be gained from **debug dialer packet**, but the **debug dialer packet** command is resource-intensive and should not be used on a busy system that has multiple dialer interfaces operating.

The following line of **debug** dialer events output for an IP packet lists the name of the DDR interface and the source and destination addresses of the packet:

Dialing cause: Async1: ip (s=172.16.1.111 d=172.16.2.22)

If this does not occur, the most common reason is improper configuration, either of the interesting traffic definitions, the state of the dialer interface, or the routing (Table 17-6).

Table 17-6 Traffic Does Not Initiate a Dial Attempt

Possible Causes	Suggested Actions
Missing or incorrect "interesting traffic" definitions	1. Using the command **show running-config**, ensure that the interface is configured with a dialer group and that there is a global level dialer list configured with a matching number. 2. Ensure that the **dialer-list** command is configured to permit either an entire protocol or to permit traffic matching an access list. 3. Verify that the access list declares packets going across the link to be interesting. One useful test is to use the privileged exec command **debug ip packet [list number]** using the number of the pertinent access list, and then attempt to **ping** or otherwise send traffic across the link. If the interesting traffic filters have been properly defined, you will see the packets in the **debug** output. If there is no **debug** output from this test, then the access list is not matching the packets.
Interface state	Using the command **show interfaces [*interface name*]**, ensure that the interface is in the state "up/up (spoofing)."
Interface in "standby" mode	Another (primary) interface on the router has been configured to use the dialer interface as a backup interface. Furthermore, the primary interface is not in a state of "down/down," which is required to bring the dialer interface out of standby mode. Also, a *backup delay* must be configured on the primary interface, or the **backup interface** command will never be enforced. To check that the dialer interface will change from standby to up/up (spoofing), it is usually necessary to pull the cable from the primary interface. Simply shutting down the primary interface with the configuration command **shutdown** will not put the primary interface into down/down, but instead will put it into administratively down, which is not the same thing. In addition, if the primary connection is via Frame Relay, the Frame Relay configuration must be done on a point-to-point serial subinterface, and the telco must be passing the "active" bit, a practice also known as "end-to-end LMI."

Table 17-6 Traffic Does Not Initiate a Dial Attempt (continued)

Possible Causes	Suggested Actions
Interface that is "administratively down"	The dialer interface has been configured with the command **shutdown**. This is also the default state of any interface when a Cisco router is booted for the very first time. Use the interface configuration command **no shutdown** to remove this impediment.
Incorrect routing	Issue the exec command **show ip route [*a.b.c.d*]**, where ***a.b.c.d*** is the address of the dialer interface of the remote router. (If **ip unnumbered** is used on the remote router, use the address of the interface listed in the **ip unnumbered** command.) The output should show a route to the remote address via the dialer interface. If there is no route, ensure that static or floating static routes have been configured by examining the output of **show running-config**. If there is a route via an interface other than the dialer interface, the implication is that DDR is being used as a backup. Examine the router configuration to make sure that static or floating static routes have been configured. The surest way to test the routing in this case is to disable the primary connection and then execute the **show ip route [*a.b.c.d*]** command to verify that the proper route has been installed in the routing table. **Note:** If you attempt this during live network operations, a dial event may be triggered. This sort of testing is best accomplished during scheduled maintenance cycles.

Placing the Call

If the routing and the interesting traffic filters are correct, a call should be initiated. This can be seen by using **debug dialer events**:

Async1 DDR: Dialing cause ip (s=10.0.0.1, d=10.0.0.2)

Async1 DDR: Attempting to dial 5551212

If the dialing cause is seen but no attempt is made to dial, the usual reason is a misconfigured dialer map or dialer profile (Table 17-7).

Table 17-7 Call Not Placed

Possible Problem	Suggested Actions
Misconfigured dialer map	Use the command **show running-config** to ensure that the dialing interface is configured with at least one **dialer map** statement that points to the protocol address and called number of the remote site.
Misconfigured dialer profile	Use the command **show running-config** to ensure that the dialer interface is configured with a **dialer pool X** command and that a dialer interface on the router is configured with a matching **dialer pool—member X**. If dialer profiles are not properly configured, you may see a **debug** message such as "Dialer1: Can't place call, no dialer pool set." Make sure that a dialer string is configured.

Async Outbound Calling—Verify Chat Script Operation

If the outbound call is a modem call, a chat script must execute for the call to proceed. For dialer map-based DDR, the chat script is invoked by the **modem-script** parameter in a dialer map command. If the DDR is dialer profile-based, this is accomplished by the command **script dialer,** configured on the TTY line. Both uses rely on a chat script existing in the router's global configuration, such as this, for example:

chat-script callout AT OK atdt\T TIMEOUT 60 CONNECT \c

In either event, the command to view the chat script activity is **debug chat**. If the dial string (such as phone number) used in the **dialer map** or **dialer string** command were 5551212, the debug output would look like this:

```
CHAT1: Attempting async line dialer script
CHAT1: Dialing using Modem script: callout & System script: none
CHAT1: process started
CHAT1: Asserting DTR
CHAT1: Chat script callout started
CHAT1: Sending string: AT
CHAT1: Expecting string: OK
CHAT1: Completed match for expect: OK
CHAT1: Sending string: atdt5551212
CHAT1: Expecting string: CONNECT
CHAT1: Completed match for expect: CONNECT
CHAT1: Chat script callout finished, status = Success
```

Chat script problems can be broken into three categories:

- Configuration error
- Modem failure
- Connection failure

Table 17-8 shows chat script failures and suggested actions.

Table 17-8 Chat Script Failure

Output from debug chat Shows:	Suggested Action
No matching chat script found for [number]	A chat script has not been configured. Add one.
Chat script dialout finished, status = Connection timed out; remote host not responding	The modem is not responding to the chat script. Verify communication with the modem (see Table 16-2 in Chapter 16).
Timeout expecting: CONNECT	**Possibility 1**: The local modem is not actually placing the call. Verify that the modem can place a call by using reverse Telnet to the modem and manually initiating a dial. **Possibility 2**: The remote modem is not answering. Test this by dialing the remote modem with an ordinary POTS telephone.
Timeout expecting: CONNECT *(continued)*	**Possibility 3**: The number being dialed is incorrect. Verify the number by dialing it manually, and correct the configuration, if necessary. **Possibility 4**: The modem trainup is taking too long or the TIMEOUT value is too low. If the local modem is external, turn up the modem speaker volume and listen to the trainup tones. If the trainup is abruptly cut off, try increasing the TIMEOUT value in the **chat-script** command. If the TIMEOUT is already 60 seconds or more, see the section "Modem Trainup," earlier in this chapter.

ISDN Outbound Calling

The first thing to check at the first suspicion of an ISDN failure, either on a BRI or a PRI, is the output from **show isdn status**. The key things to note are that Layer 1 should be Active and Layer 2 should be in a state of MULTIPLE_FRAME_ESTABLISHED. See the section "Interpreting **show isdn status** Output," in Chapter 16, for information on reading this output and for corrective measures.

For outbound ISDN calls, **debug isdn q931** and **debug isdn events** are the best tools to use. Fortunately, debugging outbound calls is very similar to debugging incoming calls. A normal successful call might look like this:

```
*Mar 20 21:07:45.025: ISDN BR0: Event: Call to 5553759 at 64 Kb/s
*Mar 20 21:07:45.033: ISDN BR0: TX ->  SETUP pd = 8  callref = 0x2C
*Mar 20 21:07:45.037:         Bearer Capability i = 0x8890
*Mar 20 21:07:45.041:         Channel ID i = 0x83
*Mar 20 21:07:45.041:         Keypad Facility i = 0x35353533373539
*Mar 20 21:07:45.141: ISDN BR0: RX <-  CALL_PROC pd = 8  callref = 0xAC
*Mar 20 21:07:45.145:         Channel ID i = 0x89
*Mar 20 21:07:45.157: ISDN BR0: received HOST_PROCEEDING
        Channel ID i = 0x0101
*Mar 20 21:07:45.161:   -------------------
        Channel ID i = 0x89
*Mar 20 21:07:45.313: ISDN BR0: RX <- CONNECT pd = 8  callref = 0xAC
*Mar 20 21:07:45.325: ISDN BR0: received HOST_CONNECT
```

Note that the CONNECT message is the key indicator of success. If a CONNECT is not received, you may see a DISCONNECT or a RELEASE_COMP ("release complete") message followed by a cause code:

***Mar 20 22:11:03.212: ISDN BR0: RX <- RELEASE_COMP pd = 8 callref = 0x8F**

***Mar 20 22:11:03.216:** Cause i = 0x8295 - Call rejected

The cause value indicates two things. The second byte of the 4- or 6-byte value indicates from where in the end-to-end call path the DISCONNECT or RELEASE_COMP was received. This can help you to localize the problem. The third and fourth bytes indicate the actual reason for the failure. See the tables that follow for the meanings of the different values.

Note: If you see this line, the likely reason is a higher-protocol failure:

Cause i = 0x8090 - Normal call clearing

PPP authentication failure is a typical reason. Turn on **debug ppp negotiation** and **debug ppp authentication** before assuming that the connection failure is necessarily an ISDN problem.

Cause Code Fields

Table 17-9 lists the ISDN cause code fields that display in the following format within the **debug** commands:

i=0x *y1 y2 z1 z2* [*a1 a2*]

Table 17-9 ISDN Cause Code Fields

Field	Value—Description
0x	The values that follow are in hexadecimal.
y1	8-ITU-T standard coding.

Table 17-9 ISDN Cause Code Fields

y2	0-User 1-Private network serving local user 2-Public network serving local user 3-Transit network 4-Public network serving remote user 5-Private network serving remote user 7-International network A-Network beyond internetworking point
z1	Class (the more significant hexadecimal number) of cause value. Refer to Table 17-10 for detailed information about possible values.
z2	Value (the less significant hexadecimal number) of cause value. Refer to Table 17-10 for detailed information about possible values.
a1	(Optional) Diagnostic field that is always 8.
a2	(Optional) Diagnostic field that is one of the following values: 0-Unknown 1-Permanent 2-Transient

Cause Values

Table 17-10 lists descriptions of some of the most commonly seen cause values of the cause information element—the third and fourth bytes of the cause code.

Table 17-10 ISDN Cause Values

Hex Value	Cause	Explanation
81	Unallocated (unassigned) number	The ISDN number was sent to the switch in the correct format; however, the number is not assigned to any destination equipment.
90	Normal call clearing	Normal call clearing has occurred.
91	User busy	The called system acknowledges the connection request but is incapable of accepting the call because all B channels are in use.
92	No user responding	The connection cannot be completed because the destination does not respond to the call.
93	No answer from user (user alerted)	The destination responds to the connection request but fails to complete the connection within the prescribed time. The problem is at the remote end of the connection.

Table 17-10 ISDN Cause Values (continued)

Hex Value	Cause	Explanation
95	Call rejected	The destination is capable of accepting the call, but it rejected the call for an unknown reason.
9C	Invalid number format	The connection could not be established because the destination address was presented in an unrecognizable format or because the destination address was incomplete.
9F	Normal, unspecified	This reports the occurrence of a normal event when no standard cause applies. No action is required.
A2	No circuit/channel available	The connection cannot be established because no appropriate channel is available to take the call.
A6	Network out of order	The destination cannot be reached because the network is not functioning correctly, and the condition might last for an extended period of time. An immediate reconnect attempt will probably be unsuccessful.
AC	Requested circuit/channel not available	The remote equipment cannot provide the requested channel for an unknown reason. This might be a temporary problem.
B2	Requested facility not subscribed	The remote equipment supports the requested supplementary service by subscription only. This is frequently a reference to long-distance service.
B9	Bearer capability not authorized	The user requested a bearer capability that the network provides, but the user is not authorized to use it. This might be a subscription problem.
D8	Incompatible destination	This indicates that an attempt was made to connect to non-ISDN equipment—for example, to an analog line.
E0	Mandatory information element missing	The receiving equipment received a message that did not include one of the mandatory information elements. This is usually the result of a D-channel error. If this error occurs systematically, report it to your ISDN service provider.
E4	Invalid information element contents	The remote equipment received a message that includes invalid information in the information element. This is usually the result of a D-channel error.

continues

For more complete information about ISDN codes and values, refer to the "ISDN Switch Codes and Values" chapter in the *Cisco IOS Debug Command Reference* for your version of IOS, in print or online at www.cisco.com/univercd/home/home.htm.

CAS Outbound Calling

For outbound calling via CAS T1 or E1 and integrated digital modems, much of the troubleshooting is similar to other DDR troubleshooting. (The same holds true for outbound integrated modem calls over a PRI line.) The unique features involved in making a call in this manner require special debugging in the event of a call failure.

As for other DDR situations, you must ensure that a call attempt is demanded—**debug dialer events** is useful for this (see the section "Verifying Dialer Operation," earlier in this chapter).

Before a call can be placed, a modem must be allocated for the call. To view this process and the subsequent call use the following **debug** commands:

debug modem

debug modem csm

debug cas

Note: The **debug cas** command first appeared in IOS version 12.0(7)T for the AS5200 and AS5300. Earlier versions of IOS use a system-level configuration command **service internal** along with the exec command **modem-mgmt debug rbs**:

Turning On the Debugs

```
router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
router(config)#service internal
router(config)#^Z

router#modem-mgmt csm ?
  debug-rbs     enable rbs debugging
  no-debug-rbs  disable rbs debugging

router#modem-mgmt csm debug-rbs
router#
neat msg at slot 0: debug-rbs is on
neat msg at slot 0: special debug-rbs is on
```

Turning Off the Debugs

```
router#
router#modem-mgmt csm no-debug-rbs
neat msg at slot 0: debug-rbs is off
```

Debugging this information on an AS5800 requires connecting to the trunk card itself.

The following is an example of a normal outbound call over a CAS T1 provisioned and configured for FXS-Ground-Start:

```
Mica Modem(1/0): Rcvd Dial String(5551111) [Modem receives digits from chat script]
CSM_PROC_IDLE: CSM_EVENT_MODEM_OFFHOOK at slot 1, port 0
CSM_RX_CAS_EVENT_FROM_NEAT:(A003):  EVENT_CHANNEL_LOCK at slot 1 and port 0
CSM_PROC_OC4_DIALING: CSM_EVENT_DSX0_BCHAN_ASSIGNED at slot 1, port 0
Mica Modem(1/0): Configure(0x1)
Mica Modem(1/0): Configure(0x2)
Mica Modem(1/0): Configure(0x5)
Mica Modem(1/0): Call Setup

neat msg at slot 0: (0/2): Tx RING_GROUND
Mica Modem(1/0): State Transition to Call Setup

neat msg at slot 0: (0/2): Rx TIP_GROUND_NORING [Telco switch goes OFFHOOK]
CSM_RX_CAS_EVENT_FROM_NEAT:(A003):  EVENT_START_TX_TONE at slot 1 and port 0
CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_DSX0_START_TX_TONE at slot 1, port 0

neat msg at slot 0: (0/2): Tx LOOP_CLOSURE [Now the router goes OFFHOOK]

Mica Modem(1/0): Rcvd Tone detected(2)
Mica Modem(1/0): Generate digits:called_party_num=5551111 len=8
Mica Modem(1/0): Rcvd Digits Generated
CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_ADDR_INFO_COLLECTED at slot 1, port 0
CSM_RX_CAS_EVENT_FROM_NEAT:(A003):  EVENT_CHANNEL_CONNECTED at slot 1 and port 0
CSM_PROC_OC5_WAIT_FOR_CARRIER: CSM_EVENT_DSX0_CONNECTED at slot 1, port 0
Mica Modem(1/0): Link Initiate
Mica Modem(1/0): State Transition to Connect
Mica Modem(1/0): State Transition to Link
Mica Modem(1/0): State Transition to Trainup
Mica Modem(1/0): State Transition to EC Negotiating
Mica Modem(1/0): State Transition to Steady State
Mica Modem(1/0): State Transition to Steady State Speedshifting
Mica Modem(1/0): State Transition to Steady State
```

debugs for T1s and E1s with other signaling types are similar.

Getting to this point in the debugging indicates that the calling and answering modems have trained and connected and that higher-layer protocols can begin to negotiate. If a modem is properly allocated for the outbound call, but the connection fails to get this far, the T1 must be examined. See Chapter 15 for T1 troubleshooting.

Troubleshooting PPP

Troubleshooting the PPP portion of a connection begins when you have established that the dial connection, ISDN or async, successfully establishes.

It is important to understand what a successful **debug** PPP sequence looks like before you troubleshoot PPP negotiation. In this way, comparing a faulty PPP **debug** session against a successfully completed **debug** PPP sequence saves you time and effort.

Following is an example of a successful PPP sequence. See Table 17-11 for a detailed description of the output fields.

```
Montecito#
Mar 13 10:57:13.415: %LINK-3-UPDOWN: Interface Async1, changed state to up
Mar 13 10:57:15.415: As1 LCP: O CONFREQ [ACKrcvd] id 2 len 25
Mar 13 10:57:15.415: As1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Mar 13 10:57:15.415: As1 LCP:    AuthProto CHAP (0x0305C22305)
Mar 13 10:57:15.415: As1 LCP:    MagicNumber 0x1084F0A2 (0x05061084F0A2)
Mar 13 10:57:15.415: As1 LCP:    PFC (0x0702)
Mar 13 10:57:15.415: As1 LCP:    ACFC (0x0802)
Mar 13 10:57:15.543: As1 LCP: I CONFACK [REQsent] id 2 len 25
Mar 13 10:57:15.543: As1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Mar 13 10:57:15.543: As1 LCP:    AuthProto CHAP (0x0305C22305)
Mar 13 10:57:15.543: As1 LCP:    MagicNumber 0x1084F0A2 (0x05061084F0A2)
Mar 13 10:57:15.543: As1 LCP:    PFC (0x0702)
Mar 13 10:57:15.547: As1 LCP:    ACFC (0x0802)
Mar 13 10:57:16.919: As1 LCP: I CONFREQ [ACKrcvd] id 4 len 23
Mar 13 10:57:16.919: As1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Mar 13 10:57:16.919: As1 LCP:    MagicNumber 0x001327B0 (0x0506001327B0)
Mar 13 10:57:16.919: As1 LCP:    PFC (0x0702)
Mar 13 10:57:16.919: As1 LCP:    ACFC (0x0802)
Mar 13 10:57:16.919: As1 LCP:    Callback 6  (0x0D0306)
Mar 13 10:57:16.919: As1 LCP: O CONFREJ [ACKrcvd] id 4 len 7
Mar 13 10:57:16.919: As1 LCP:    Callback 6  (0x0D0306)
Mar 13 10:57:17.047: As1 LCP: I CONFREQ [ACKrcvd] id 5 len 20
Mar 13 10:57:17.047: As1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Mar 13 10:57:17.047: As1 LCP:    MagicNumber 0x001327B0 (0x0506001327B0)
Mar 13 10:57:17.047: As1 LCP:    PFC (0x0702)
Mar 13 10:57:17.047: As1 LCP:    ACFC (0x0802)
Mar 13 10:57:17.047: As1 LCP: O CONFACK [ACKrcvd] id 5 len 20
Mar 13 10:57:17.047: As1 LCP:    ACCM 0x000A0000 (0x0206000A0000)
Mar 13 10:57:17.047: As1 LCP:    MagicNumber 0x001327B0 (0x0506001327B0)
Mar 13 10:57:17.047: As1 LCP:    PFC (0x0702)
Mar 13 10:57:17.047: As1 LCP:    ACFC (0x0802)
Mar 13 10:57:17.047: As1 LCP: State is Open
Mar 13 10:57:17.047: As1 PPP: Phase is AUTHENTICATING, by this end
Mar 13 10:57:17.047: As1 CHAP: O CHALLENGE id 1 len 28 from "Montecito"
Mar 13 10:57:17.191: As1 CHAP: I RESPONSE id 1 len 30 from "Goleta"
Mar 13 10:57:17.191: As1 CHAP: O SUCCESS id 1 len 4
Mar 13 10:57:17.191: As1 PPP: Phase is UP
Mar 13 10:57:17.191: As1 IPCP: O CONFREQ [Closed] id 1 len 10
Mar 13 10:57:17.191: As1 IPCP:    Address 172.22.66.23 (0x0306AC164217)
Mar 13 10:57:17.303: As1 IPCP: I CONFREQ [REQsent] id 1 len 40
Mar 13 10:57:17.303: As1 IPCP:    CompressType VJ 15 slots CompressSlotID (0x0206002D0F01)
Mar 13 10:57:17.303: As1 IPCP:    Address 0.0.0.0 (0x030600000000)
Mar 13 10:57:17.303: As1 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Mar 13 10:57:17.303: As1 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Mar 13 10:57:17.303: As1 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Mar 13 10:57:17.303: As1 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Mar 13 10:57:17.303: As1 IPCP: O CONFREJ [REQsent] id 1 len 22
Mar 13 10:57:17.303: As1 IPCP:    CompressType VJ 15 slots CompressSlotID (0x0206002D0F01)
Mar 13 10:57:17.303: As1 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Mar 13 10:57:17.303: As1 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Mar 13 10:57:17.319: As1 CCP: I CONFREQ [Not negotiated] id 1 len 15
Mar 13 10:57:17.319: As1 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
Mar 13 10:57:17.319: As1 CCP:    Stacker history 1 check mode EXTENDED (0x1105000104)
Mar 13 10:57:17.319: As1 LCP: O PROTREJ [Open] id 3 len 21 protocol CCP
Mar 13 10:57:17.319: As1 LCP:  (0x80FD0101000F12060000000111050001)
Mar 13 10:57:17.319: As1 LCP:  (0x04)
Mar 13 10:57:17.319: As1 IPCP: I CONFACK [REQsent] id 1 len 10
Mar 13 10:57:17.319: As1 IPCP:    Address 172.22.66.23 (0x0306AC164217)
Mar 13 10:57:18.191: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async1, changed state
to up
```

```
Mar 13 10:57:19.191: As1 IPCP: TIMEout: State ACKrcvd
Mar 13 10:57:19.191: As1 IPCP: O CONFREQ [ACKrcvd] id 2 len 10
Mar 13 10:57:19.191: As1 IPCP:    Address 172.22.66.23 (0x0306AC164217)
Mar 13 10:57:19.315: As1 IPCP: I CONFACK [REQsent] id 2 len 10
Mar 13 10:57:19.315: As1 IPCP:    Address 172.22.66.23 (0x0306AC164217)
Mar 13 10:57:20.307: As1 IPCP: I CONFREQ [ACKrcvd] id 2 len 34
Mar 13 10:57:20.307: As1 IPCP:    Address 0.0.0.0 (0x030600000000)
Mar 13 10:57:20.307: As1 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Mar 13 10:57:20.307: As1 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Mar 13 10:57:20.307: As1 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Mar 13 10:57:20.307: As1 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Mar 13 10:57:20.307: As1 IPCP: O CONFREJ [ACKrcvd] id 2 len 16
Mar 13 10:57:20.307: As1 IPCP:    PrimaryWINS 0.0.0.0 (0x820600000000)
Mar 13 10:57:20.307: As1 IPCP:    SecondaryWINS 0.0.0.0 (0x840600000000)
Mar 13 10:57:20.419: As1 IPCP: I CONFREQ [ACKrcvd] id 3 len 22
Mar 13 10:57:20.419: As1 IPCP:    Address 0.0.0.0 (0x030600000000)
Mar 13 10:57:20.419: As1 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
Mar 13 10:57:20.419: As1 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
Mar 13 10:57:20.419: As1 IPCP: O CONFNAK [ACKrcvd] id 3 len 22
Mar 13 10:57:20.419: As1 IPCP:    Address 10.1.1.1 (0x03060A010101)
Mar 13 10:57:20.419: As1 IPCP:    PrimaryDNS 171.68.10.70 (0x8106AB440A46)
Mar 13 10:57:20.419: As1 IPCP:    SecondaryDNS 171.68.10.140 (0x8306AB440A8C)
Mar 13 10:57:20.543: As1 IPCP: I CONFREQ [ACKrcvd] id 4 len 22
Mar 13 10:57:20.543: As1 IPCP:    Address 10.1.1.1 (0x03060A010101)
Mar 13 10:57:20.547: As1 IPCP:    PrimaryDNS 171.68.10.70 (0x8106AB440A46)
Mar 13 10:57:20.547: As1 IPCP:    SecondaryDNS 171.68.10.140 (0x8306AB440A8C)
Mar 13 10:57:20.547: As1 IPCP: O CONFACK [ACKrcvd] id 4 len 22
Mar 13 10:57:20.547: As1 IPCP:    Address 10.1.1.1 (0x03060A010101)
Mar 13 10:57:20.547: As1 IPCP:    PrimaryDNS 171.68.10.70 (0x8106AB440A46)
Mar 13 10:57:20.547: As1 IPCP:    SecondaryDNS 171.68.10.140 (0x8306AB440A8C)
Mar 13 10:57:20.547: As1 IPCP: State is Open
Mar 13 10:57:20.551: As1 IPCP: Install route to 10.1.1.1
```

Note that your debugs may appear in a different format. This output is the newer PPP debugging output format, which was modified in IOS version 11.2(8). See Chapter 16 for an example of PPP debugging with the older versions of IOS.

Table 17-11 PPP LCP Negotiation Details

Time Stamp	Description
10:57:15.415	Outgoing configuration request (O CONFREQ). The NAS sends an outgoing PPP configuration request packet to the client.
10:57:15.543	Incoming configuration acknowledgment (I CONFACK). The client acknowledges Montecito's PPP request.
10:57:16.919	Incoming configuration request (I CONFREQ). The client wants to negotiate the callback protocol.
10:57:16.919	Outgoing configuration reject (O CONFREJ). The NAS rejects the callback option.
10:57:17.047	Incoming configuration request (I CONFREQ). The client requests a new set of options. Notice that Microsoft Callback is not requested this time.
10:57:17.047	Outgoing configuration acknowledgment (O CONFACK). The NAS accepts the new set of options.
10:57:17.047	PPP LCP negotiation is completed successfully (LCP: State is Open). Both sides have acknowledged (CONFACK) the other side's configuration request (CONFREQ).

Table 17-11 PPP LCP Negotiation Details (continued)

Time Stamp	Description
10:57:17.047 to 10:57:17.191	PPP authentication is completed successfully. After LCP negotiates, authentication starts. Authentication must take place before any network protocols, such as IP, are delivered. Both sides authenticate with the method negotiated during LCP. Montecito is authenticating the client using CHAP.
10:57:20.551	The state is open for IP Control Protocol (IPCP). A route is negotiated and installed for the IPCP peer, which is assigned IP address 1.1.1.1.

Link Control Protocol

Two types of problems are typically encountered during LCP negotiation. The first occurs when one peer makes configuration requests that the other peer cannot or will not acknowledge. Although this is a frequent occurrence, it can be a problem if the requester insists on the parameter. A typical example is when negotiating AUTHTYPE. For instance, many access servers are configured to accept only CHAP for authentication. If the caller is configured to do only PAP authentication, CONFREQs and CONFNAKs will be exchanged until one peer drops the connection.

```
BR0:1 LCP: I CONFREQ [ACKrcvd] id 66 len 14
BR0:1 LCP:    AuthProto PAP (0x0304C023)
BR0:1 LCP:    MagicNumber 0xBC6B9F91 (0x0506BC6B9F91)
BR0:1 LCP: O CONFNAK [ACKrcvd] id 66 len 9
BR0:1 LCP:    AuthProto CHAP (0x0305C22305)
BR0:1 LCP: I CONFREQ [ACKrcvd] id 67 len 14
BR0:1 LCP:    AuthProto PAP (0x0304C023)
BR0:1 LCP:    MagicNumber 0xBC6B9F91 (0x0506BC6B9F91)
BR0:1 LCP: O CONFNAK [ACKrcvd] id 67 len 9
BR0:1 LCP:    AuthProto CHAP (0x0305C22305)
BR0:1 LCP: I CONFREQ [ACKrcvd] id 68 len 14
BR0:1 LCP:    AuthProto PAP (0x0304C023)
BR0:1 LCP:    MagicNumber 0xBC6B9F91 (0x0506BC6B9F91)
BR0:1 LCP: O CONFNAK [ACKrcvd] id 68 len 9
BR0:1 LCP:    AuthProto CHAP (0x0305C22305)
```

. . . and so on.

The second type of problem seen in LCP is when *only* outbound CONFREQs are seen on one or both peers, as in the example that follows. This is usually the result of what is referred to as a *speed mismatch* at the lower layer. This condition can occur in either async or ISDN DDR.

LCP failure example:

```
Jun 10 19:57:59.768: As5 PPP: Phase is ESTABLISHING, Active Open
Jun 10 19:57:59.768: As5 LCP: O CONFREQ [Closed] id 64 len 25
Jun 10 19:57:59.768: As5 LCP: ACCM 0x000A0000 (0x0206000A0000)
Jun 10 19:57:59.768: As5 LCP: AuthProto CHAP (0x0305C22305)
Jun 10 19:57:59.768: As5 LCP: MagicNumber 0x5779D9D2 (0x05065779D9D2)
Jun 10 19:57:59.768: As5 LCP: PFC (0x0702)
Jun 10 19:57:59.768: As5 LCP: ACFC (0x0802)
Jun 10 19:58:01.768: As5 LCP: TIMEout: State REQsent
Jun 10 19:58:01.768: As5 LCP: O CONFREQ [REQsent] id 65 len 25
Jun 10 19:58:01.768: As5 LCP: ACCM 0x000A0000 (0x0206000A0000)
Jun 10 19:58:01.768: As5 LCP: AuthProto CHAP (0x0305C22305)
Jun 10 19:58:01.768: As5 LCP: MagicNumber 0x5779D9D2 (0x05065779D9D2)
Jun 10 19:58:01.768: As5 LCP: PFC (0x0702)
Jun 10 19:58:01.768: As5 LCP: ACFC (0x0802).
Jun 10 19:58:03.768: As5 LCP: TIMEout: State REQsent
Jun 10 19:58:03.768: As5 LCP: O CONFREQ [REQsent] id 66 len 25
Jun 10 19:58:03.768: As5 LCP: ACCM 0x000A0000 (0x0206000A0000)
Jun 10 19:58:03.768: As5 LCP: AuthProto CHAP (0x0305C22305)
Jun 10 19:58:03.768: As5 LCP: MagicNumber 0x5779D9D2 (0x05065779D9D2)
Jun 10 19:58:03.768: As5 LCP: PFC (0x0702)
Jun 10 19:58:03.768: As5 LCP: ACF.C (0x0802)
Jun 10 19:58:05.768: As5 LCP: TIMEout: State REQsent
Jun 10 19:58:05.768: As5 LCP: O CONFREQ [REQsent] id 67 len 25
```

This repeats every 2 seconds until this occurs:

```
Jun 10 19:58:19.768: As5 LCP: O CONFREQ [REQsent] id 74 len 25
Jun 10 19:58:19.768: As5 LCP: ACCM 0x000A0000 (0x0206000A0000)
Jun 10 19:58:19.768: As5 LCP: AuthProto CHAP (0x0305C22305)
Jun 10 19:58:19.768: As5 LCP: MagicNumber 0x5779D9D2 (0x05065779D9D2)
Jun 10 19:58:19.768: As5 LCP: PFC (0x0702)
Jun 10 19:58:19.768: As5 LCP: ACFC (0x0802)
Jun 10 19:58:21.768: As5 LCP: TIMEout: State REQsent
Jun 10 19:58:21.768: TTY5: Async Int reset: Dropping DTR
```

If the connection is asynchronous, the probable cause is a speed mismatch between the router and its modem, usually as a result of having failed to lock the DTE speed of the modem to the configured speed of the TTY line. The problem may be found on either or both of the peers; both should be checked. (See Table 17-1.)

If the symptoms are seen when the connection is over ISDN, the problem likely is that one peer is connecting at 56K, while the other is at 64K. Although this condition is rare, it does happen; the problem could be one or both peers, or possibly the telco itself. Use **debug isdn q931** and examine the SETUP messages on each of the peers. The bearer capability sent from one peer should match the bearer capability seen in the SETUP message received on the other peer. As a possible remedy, you can configure the dialing speed, 56K or 64K, in either the interface-level command **dialer map** or in the command **dialer isdn speed** configured under a map class.

```
*Mar 20 21:07:45.033: ISDN BR0: TX ->  SETUP pd = 8  callref = 0x2C
*Mar 20 21:07:45.037:         Bearer Capability i = 0x8890
*Mar 20 21:07:45.041:         Channel ID i = 0x83
*Mar 20 21:07:45.041:         Keypad Facility i = 0x35353533373539
```

This situation is one that may warrant a call to the Cisco TAC. Collect the following outputs from both peers before calling the TAC:

show running-config

show version

debug isdn q931

debug isdn events

debug ppp negotiation

Authentication

Failed authentication is the single most common reason for a PPP failure. Misconfigured or mismatched usernames and passwords create error messages in **debug** output.

The following example shows that the username Goleta does not have permission to dial in to the NAS, which does not have a local username configured for this user. To fix the problem, use the **username** *name* **password** *password* command to add the username Goleta to the NAS's local AAA database:

```
Mar 13 11:01:42.399: As2 LCP: State is Open
Mar 13 11:01:42.399: As2 PPP: Phase is AUTHENTICATING, by this end
Mar 13 11:01:42.399: As2 CHAP: O CHALLENGE id 1 len 28 from "Montecito"
Mar 13 11:01:42.539: As2 CHAP: I RESPONSE id 1 len 30 from "Goleta"
Mar 13 11:01:42.539: As2 CHAP: Unable to validate Response.  Username Goleta not found
Mar 13 11:01:42.539: As2 CHAP: O FAILURE id 1 len 26 msg is "Authentication failure"
Mar 13 11:01:42.539: As2 PPP: Phase is TERMINATING
```

The following example shows that the username Goleta is configured on the NAS. However, the password comparison failed. To fix this problem, use the **username** *name* **password** *password* command to specify the correct login password for Goleta:

```
Mar 13 11:04:06.843: As3 LCP: State is Open
Mar 13 11:04:06.843: As3 PPP: Phase is AUTHENTICATING, by this end
Mar 13 11:04:06.843: As3 CHAP: O CHALLENGE id 1 len 28 from "Montecito"
Mar 13 11:04:06.987: As3 CHAP: I RESPONSE id 1 len 30 from "Goleta"
Mar 13 11:04:06.987: As3 CHAP: O FAILURE id 1 len 25 msg is "MD/DES compare failed"
Mar 13 11:04:06.987: As3 PPP: Phase is TERMINATING
```

Network Control Protocol

After the peers have successfully performed whatever authentication is required, the negotiation moves into the NCP phase. If both peers are properly configured, the NCP negotiation might look like the following example, which shows a client PC dialing into and negotiating with a NAS:

```
solvang# show debug
     Generic IP:
     IP peer address activity debugging is on
     PPP:
     PPP protocol negotiation debugging is on

*Mar  1 21:35:04.186: As4 PPP: Phase is UP
*Mar  1 21:35:04.190: As4 IPCP: O CONFREQ [Not negotiated] id 1 len 10
*Mar  1 21:35:04.194: As4 IPCP:    Address 10.1.2.1 (0x03060A010201)
*Mar  1 21:35:04.282: As4 IPCP: I CONFREQ [REQsent] id 1 len 28
*Mar  1 21:35:04.282: As4 IPCP:    CompressType VJ 15 slots CompressSlotID
(0x0206002D0F01)
*Mar  1 21:35:04.286: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:04.290: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:04.298: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:04.306: As4 IPCP: O CONFREJ [REQsent] id 1 len 10
*Mar  1 21:35:04.310: As4 IPCP:    CompressType VJ 15 slots CompressSlotID
(0x0206002D0F01)
*Mar  1 21:35:04.314: As4 CCP: I CONFREQ [Not negotiated] id 1 len 15
*Mar  1 21:35:04.318: As4 CCP:    MS-PPC supported bits 0x00000001 (0x120600000001)
*Mar  1 21:35:04.318: As4 CCP:    Stacker history 1 check mode EXTENDED (0x1105000104)
*Mar  1 21:35:04.322: As4 LCP: O PROTREJ [Open] id 3 len 21 protocol CCP
*Mar  1 21:35:04.326: As4 LCP:  (0x80FD0101000F12060000000111050001)
*Mar  1 21:35:04.330: As4 LCP:  (0x04)
*Mar  1 21:35:04.334: As4 IPCP: I CONFACK [REQsent] id 1 len 10
*Mar  1 21:35:04.338: As4 IPCP:    Address 10.1.2.1 (0x03060A010201)
*Mar  1 21:35:05.186: %LINEPROTO-5-UPDOWN: Line protocol on Interface Async4, changed
state to up
*Mar  1 21:35:07.274: As4 IPCP: I CONFREQ [ACKrcvd] id 2 len 22
*Mar  1 21:35:07.278: As4 IPCP:    Address 0.0.0.0 (0x030600000000)
*Mar  1 21:35:07.282: As4 IPCP:    PrimaryDNS 0.0.0.0 (0x810600000000)
*Mar  1 21:35:07.286: As4 IPCP:    SecondaryDNS 0.0.0.0 (0x830600000000)
*Mar  1 21:35:07.294: As4 IPCP: O CONFNAK [ACKrcvd] id 2 len 22
*Mar  1 21:35:07.298: As4 IPCP:    Address 10.1.2.2 (0x03060A010202)
*Mar  1 21:35:07.302: As4 IPCP:    PrimaryDNS 10.2.2.3 (0x81060A020203)
*Mar  1 21:35:07.310: As4 IPCP:    SecondaryDNS 10.2.3.1 (0x83060A020301)
*Mar  1 21:35:07.426: As4 IPCP: I CONFREQ [ACKrcvd] id 3 len 22
*Mar  1 21:35:07.430: As4 IPCP:    Address 10.1.2.2 (0x03060A010202)
*Mar  1 21:35:07.434: As4 IPCP:    PrimaryDNS 10.2.2.3 (0x81060A020203)
*Mar  1 21:35:07.442: As4 IPCP:    SecondaryDNS 10.2.3.1 (0x83060A020301)
*Mar  1 21:35:07.446: ip_get_pool: As4: validate address = 10.1.2.2
*Mar  1 21:35:07.450: ip_get_pool: As4: using pool default
*Mar  1 21:35:07.450: ip_get_pool: As4: returning address = 10.1.2.2
*Mar  1 21:35:07.454: set_ip_peer_addr: As4: address = 10.1.2.2 (3) is redundant
*Mar  1 21:35:07.458: As4 IPCP: O CONFACK [ACKrcvd] id 3 len 22
*Mar  1 21:35:07.462: As4 IPCP:    Address 10.1.2.2 (0x03060A010202)
*Mar  1 21:35:07.466: As4 IPCP:    PrimaryDNS 10.2.2.3 (0x81060A020203)
*Mar  1 21:35:07.474: As4 IPCP:    SecondaryDNS 10.2.3.1 (0x83060A020301)
*Mar  1 21:35:07.478: As4 IPCP: State is Open
*Mar  1 21:35:07.490: As4 IPCP: Install route to 10.1.2.2
```

Table 17-12 PPP NCP Negotiation Details

Time Stamp	Description
21:35:04.190	Outgoing configuration request (O CONFREQ). The NAS sends an outgoing PPP configuration request packet containing its IP address to the peer.
21:35:04.282	Incoming CONFREQ. The peer requests to do VJ header compression and needs an IP address for itself, as well as addresses of the primary and secondary DNS servers.
21:35:04.306	Outbound Config-Reject (CONFREJ). VJ header compression is rejected.
21:35:04.314 to 21:35:04.330	The peer sends a request to do Compression Control Protocol; the entire protocol is rejected by the NAS by means of a PROTREJ message. The peer should not (and does not) attempt to retry CCP.
21:35:04.334	The peer acknowledges the IP address of the NAS with a CONFACK.
21:35:07.274	Incoming CONFREQ. The peer no longer requests to do VJ header compression, but it still needs an IP address for itself, as well as addresses of the primary and secondary DNS servers.
21:35:07.294	The NAS sends a CONFNAK containing the address that it wants the peer to use, along with addresses of the primary and secondary DNS servers.
21:35:07.426	The peer sends the addresses back to the NAS—in effect, an attempt to confirm that the addresses were properly received.
21:35:07.458	The NAS acknowledges the addresses with a CONFACK.
21:35:07.478	Each side of the connection has issued a CONFACK, so negotiation is finished. The command **show interfaces Async4** on the NAS will show “IPCP: Open.”
21:35:07.490	A host route to the remote peer is installed in the NAS’s routing table.

It should be noted that it is possible for the peers to simultaneously negotiate more than one Layer 3 protocol. It is not uncommon, for instance, to see IP and IPX being negotiated. It is also possible for one protocol to successfully negotiate while the other fails to do so.

Any problems that occur during NCP negotiation can typically be traced to the configurations of the negotiating peers. If PPP negotiation fails during the NCP phase, take the steps outlined in Table 17-13.

Table 17-13 Troubleshooting NCP

Task	Steps
Verify interface protocol configuration.	Examine the output of the privileged exec command **show running-config**. Verify that the interface is configured to support the protocol that you want to run over the connection.
Verify interface address.	Confirm that the interface in question has an address configured. If your are using **ip unnumbered [*interface-name*]** or **ipx ppp-client loopback [*number*]**, ensure that the referenced interface is configured with an address.

Table 17-13 Troubleshooting NCP (continued)

Task	Steps
Verify client address availability.	If the NAS is supposed to issue an IP address to the caller, ensure that such an address is available. The IP address to be handed out to the caller can be derived in several ways: • **Configured locally on the interface**—Check the interface configuration for the command **peer default ip address a.b.c.d.** In practice, this method should be used only on interfaces that accept a connections from a single caller, such as on an async (*not* a group-async) interface. • **From an address pool locally configured on the NAS**—The interface should have the command **peer default ip address pool [*pool-name*]**. In addition, the pool must be defined at the system level with the command **ip local pool [*pool-name*] [*first-address*] [*last-address*]**. The range of addresses defined in the pool should be large enough to accommodate as many simultaneously connected callers as the NAS is capable of. (QQ17) • **With QQ17**—-The range of address cannot be greater than the maximum amount of simultaneous connection that can be handled on the routing platform, but it can be less. • **From a DHCP server**—The NAS interface must be configured with the command **peer default ip address dhcp**. Furthermore, the NAS must be configured to point to a DHCP server with the global configuration command **ip dhcp-server [*address*]**. • **Via AAA**—If you are using TACACS+ or RADIUS for authorization, the AAA server can be configured to hand a specific IP address to a given caller every time that caller connects.
Verify server address configuration.	To return the configured addresses of domain name servers or Windows NT servers in response to BOOTP requests, ensure that the global-level commands **async-bootp dns-server [*address*]** and **async-bootp nbns-server [*address*]** are configured.

Note that although the command **async-bootp subnet-mask [*mask*]** can be configured on the NAS, the subnet mask will *not* be negotiated between the NAS and a PPP dial-in client PC. Because of the nature of point-to-point connections, the client automatically uses the IP address of the NAS (learned during IPCP negotiation) as the default gateway. The subnet mask is not needed in that point-to-point environment. The PC knows that if the destination address does not match the local address, the packet should be forwarded to the default gateway (NAS), which is always reached via the PPP link.

Before Calling Cisco Systems' TAC Team

Before calling Cisco Systems' Technical Assistance Center (TAC), make sure that you have read through this chapter and completed the actions suggested for your system's problem.

Additionally, do the following and document the results so that the TAC can better assist you:

- For all problems, collect the output of **show running-config** and **show version**. Ensure that the command **service timestamps debug datetime msec** is in the configuration.
- For DDR problems, collect the following:
  - show dialer map
  - debug dialer
  - debug ppp negotiation
  - debug ppp authentication
- If ISDN is involved, collect the following:
  - show isdn status
  - debug isdn q931
  - debug isdn events
- If modems are involved, collect the following:
  - show lines
  - **show line** [*x*]
  - **show modem** (if integrated modems are involved)
  - **show modem version** (if integrated modems are involved)
  - debug modem
  - **debug modem csm** (if integrated modems are involved)
  - **debug chat** (if a DDR scenario)
- If T1s or PRIs are involved, collect:
  - show controller t1

Additional Sources

- Cisco IOS Dial Solutions Guide
- The TAC Technology Support Pages: www.cisco.com/tac/