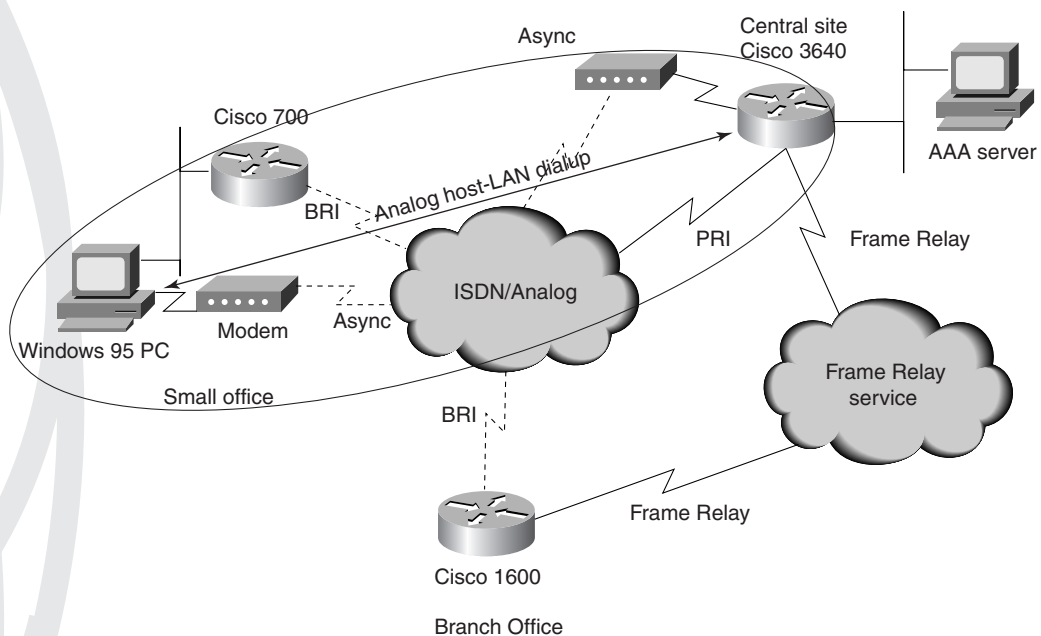


# CHAPTER 4

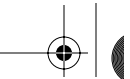
## Configuring Asynchronous Connections with Modems

In the first three chapters, you learned how to identify your communication needs and how to select the equipment. This chapter teaches you how to establish remote connections from a home office or telecommuter to a central site, as shown in Figure 4-1.

**Figure 4-1** *Small Office or Telecommuter Calling the Central Site*



You will learn how to configure an access server for modem connectivity so that telecommuters can access the Central site by using asynchronous connection. To do this, you will learn how to perform a reverse Telnet session to the modem, how to configure it for basic asynchronous operations, and how to use the modem autoconfiguration.



### Dial-Up Terminology

Following are the definitions of two terms (taken from the Cisco Web site) that you will frequently encounter in this chapter:

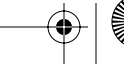
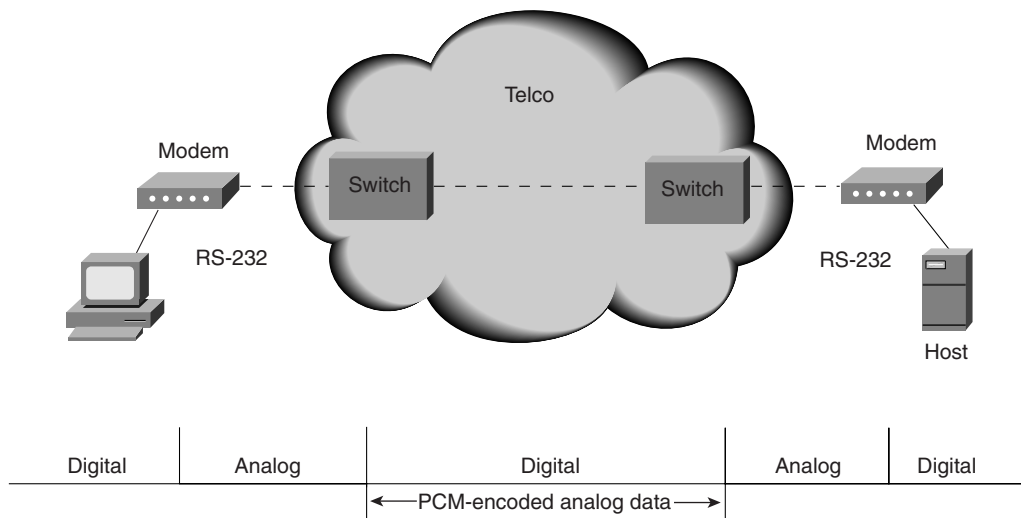
**Analog** An electrical circuit that is represented by means of continuous, variable physical quantities (such as voltages and frequencies), as opposed to discrete representations (such as the 0/1, off/on representation of digital circuits). Signal transmission over wires or through the air in which information is conveyed through variation of some combination of signal amplitude, frequency, and phase.

**Asynchronous** Term describing digital signals that are transmitted without precise clocking. Such signals generally have different frequencies and phase relationships. Asynchronous transmissions usually encapsulate individual characters in control bits (called start and stop bits) that designate the beginning and end of each character.

## Modem Overview

Regular phone lines, also known as Plain Old Telephone Service (POTS), were designed to carry voice traffic and are analog by definition. On the other hand, computers are digital and communicate digitally. For the digital equipment to communicate over an analog connection, its current (digital) needs to be transformed into a tone (analog). A *modem* (modulator/demodulator) converts digital signals to analog, and vice versa. It converts digital to analog, only to be converted back to digital, as seen in Figure 4-2. The typical maximum data rate is usually limited to 28.8 to 56 Kbps. Multiple analog to digital conversions introduce noise.

Figure 4-2 Typical Modem Connection





---

### Universal Asynchronous Receiver/Transmitter (UART)

Transmission speed can also be limited by a workstation's UART type. The UART is a computer component that handles asynchronous serial communication. Every computer contains an UART to manage serial ports, such as those used for a mouse or an external modem. Every internal modems has its own UART.

UARTs are controlled by a clock usually running at 1.84 MHz and have a maximum throughput of 115 Kbps. UARTs have a buffer to temporarily hold incoming data. This buffer varies by the model, but it is usually quite small.

Type of UARTs: 16C450

16450

16550 uses a 16-byte buffer

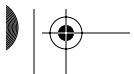
16550af

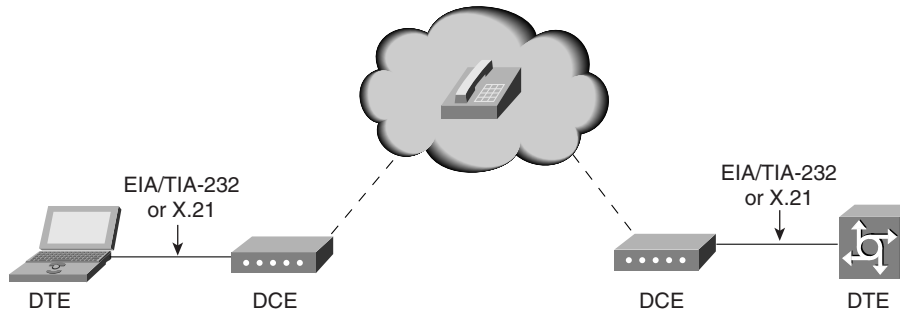
16750 uses a 64-byte transmit buffer and a 56-byte receive buffer

---

Within the telco cloud and between switches, the analog data is pulse code modulation (PCM)-encoded for transmission over telco facilities. The recovery and amplification of the digital signals reduce and eliminate the noise that would be created by subjecting analog signals to the same process. Amplifying an analog signal amplifies any noise in the signal. Amplifying a digital signal means re-creating only the on or off state of the signal.

Data terminal equipment (DTE) are end devices such as PCs, workstations, and mainframe computers. End devices communicate with each other through data communications equipment (DCE) such as modems, channel service units (CSUs), and data service units (DSUs). Figure 4-3 displays the DTE-DCE interface. DCE can also be expanded to mean data circuit-terminating equipment, which is the International Telecommunication Union-Telecommunications Standardization Sector (ITU-TSS, or simply ITU-T), formerly known as CCITT. The Electronic Industries Association (EIA) defines the DCE acronym as *data communications equipment*.



**Figure 4-3** *The DTE-DCE Interface*

- DTE = Data terminal equipment
- DCE = Data communications equipment

The EIA/TIA-232 standard defines the interface between DTE and DCE. TIA stands for Telecommunications Industries Association.

The end-to-end communication path between two DTEs consists of three segments (as illustrated in Figure 4-3): DTE-DCE, DCE-DCE, and DCE-DTE. You must administer a set of cabling and configuration elements for each segment.

**NOTE** The EIA/TIA-232-C (formerly known as RS-232-C) standard is the most commonly used asynchronous interface for data communications in North America. The RS-232 standard was first issued in 1962, and its third revision, RS-232-C, was issued in August 1969. Although the ubiquitous D-shaped 25-pin connector (DB-25) has become the market norm for EIA/TIA-232-C interfaces, it was not specified in the original RS-232-C standard. Many EIA/TIA-232-C devices use other connectors, such as the DB-9 or RJ-11/RJ-45 modular connectors. X.21 is a European standard that defines the DCE-DTE interface. For more information on these and other standards, refer to Cisco Connection Online (CCO) or another data communications reference text.

## Modem Signaling and Cabling

A number of different standards define the signaling over a serial cable, including EIA/TIA-232, X.21, V.35, EIA/TIA-449, EIA-530, and EIA-613 HSSI. Each standard defines the signals on the cable and specifies the connector at the end of the cable.

With the 25-pin connector of EIA/TIA-232 standard, only eight pins are actually used for connecting a DTE (such as an access server) to a DCE (such as a modem). The other 17 signals are not interesting and are ignored. The eight interesting signals (pins) can be grouped into three categories by their functionality:

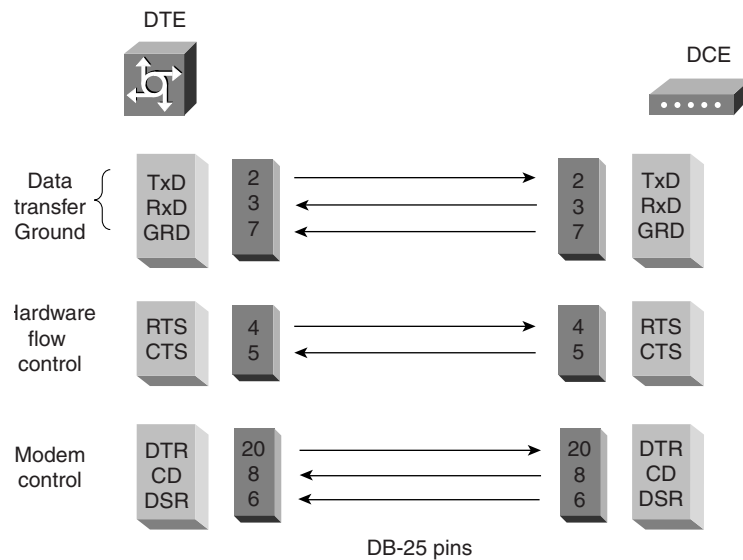
- Data transfer group
- Hardware flow control group
- Modem control group

Figure 4-4 illustrates these three groups.

### Data Transfer Group

The data transfer group signals and pin designation, also known as *pinout*, for the EIA/TIA-232 specification in Figure 4-4 are explained in Table 4-1.

**Figure 4-4** Data, Flow Control, and Modem Signaling

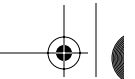


**Table 4-1** Data Transfer Group

Signal	Description
TxD	Transmit Data. The DTE transmits data to the DCE.
RxD	Receive Data. The DTE receives data from the DCE.
GRD	Ground (pin 7). Provides the ground reference for voltage measurements.

### Flow Control Group

Pins 4 and 5 form the hardware flow control group, as shown in Figure 4-4. These signals are activated between the DCE and the DTE when the equipment is ready to accept data. Table 4-2 explains the different flow control signals.



60 Chapter 4: Configuring Asynchronous Connections with Modems

**Table 4-2** *Flow Control Group*

Signal	Description
RTS	Request To Send. The DTE has buffers available to receive from the DCE. Originally intended for the DTE to request to send data during a half duplex operation (in which data can only be sent in one direction at a time), this signal is now used for full duplex communication to indicate to the DCE that the DTE is ready to receive data. This is a signal from the computer or router telling the modem when to send data.
CTS	Clear To Send. The DCE has buffers that are available to take data from the DTE. Initially used by the DCE to indicate that the DTE could transmit in half duplex mode in response to RTS. It is still used to indicate that the DTE may transmit for hardware flow control under full duplex operation. This signal is used by the modem to tell the computer when to send data.

### Modem Control Group

Finally, the remaining interesting signals between a DTE device and a DCE form the modem control group, covered in Table 4-3. These signals are used between the DTE and DCE to initiate, terminate, and monitor the status of the connection.

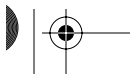
**Table 4-3** *Modem Control Group*

Signal	Description
DTR	Data terminal ready. This signal is controlled by the DTE. The DTE indicates to the DCE that the equipment (computer or router) is connected and available to receive data.
CD	Carrier Detect. This signal is controlled by the DCE, and it indicates that it has established an acceptable carrier signal with a remote DCE (there is a DCE-to-DCE connection).
DSR	Data Set Ready (pin 6). The DCE is ready for use. This pin is not used on modem connections. The DSR is active as soon as a modem is turned on.

**TIP** In the teletype days, flow control was done with inband signaling using Xon/Xoff. With higher DTE speeds and faster workstations, modems and computers were not always able to exchange this inband signaling in a timely fashion. Therefore, a set of electrical signals was developed to manage the flow control and the modem control.

### Communication Termination

Either the DTE device or the DCE device may signal for the connection to be terminated. The signals that are used for this function are DTR from the DTE or the modem recognizing the loss of the CD signal. Therefore, a modem connection can be terminated in two ways:



- **DTE initiated**—The access server or computer can drop the DTR signal. The modem must be programmed to terminate the connection on loss of DTR and restore to saved settings in its NVRAM.
- **DCE initiated**—The access server detects Carrier Detect low and terminates the connection. The modem must be programmed so that the CD reflects the state of the carrier.

When modem control is not configured properly, the following symptoms might occur:

- The modem does not hang up when you quit your session. This means the DTR is not dropped or recognized, so the modem is not aware that it should break the connection.
- You end up in someone else's session, which means that the CD is not dropped or recognized. This scenario happens when Caller A terminates its dial-up session, and the modem does not pass the true state of the CD to the DTE. The access server is not aware that Caller A terminated its session, so it maintains the line for Caller A. When a new caller, Caller B, comes in by the same line (interface), the access server continues with the session previously initiated by Caller A, instead of starting a new one. Thus, Caller B ends up in Caller A's session without having to authenticate. It is, therefore, very important that the true state of CD be always passed back to the DTE, so the access server terminates sessions when callers hang up.

## Modem Operation

Modems perform their basic operations as follows (in one direction) and as seen in Figure 4-5:

- 1 Outgoing data from an originating DTE comes into the sending modem via the TxD pin.
- 2 If the sending modem's buffer is nearly full, the modem can control flow (via hardware) by lowering the CTS signal, thus instructing the DTE to not use TxD.
- 3 The data is compressed by using a proper algorithm (MNP 5 or V.42bis) that is mutually agreed upon between the two communicating modems.
- 4 The data is then packetized, where windowing, checksum, error control (using MNP 4 or LAP-M), and retransmission are performed.

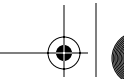
---

### NOTE

Here, the term *packetized* is not referring to an IP packet or layer-3 PDU. Rather, it refers to the preparation of the data by the modem.

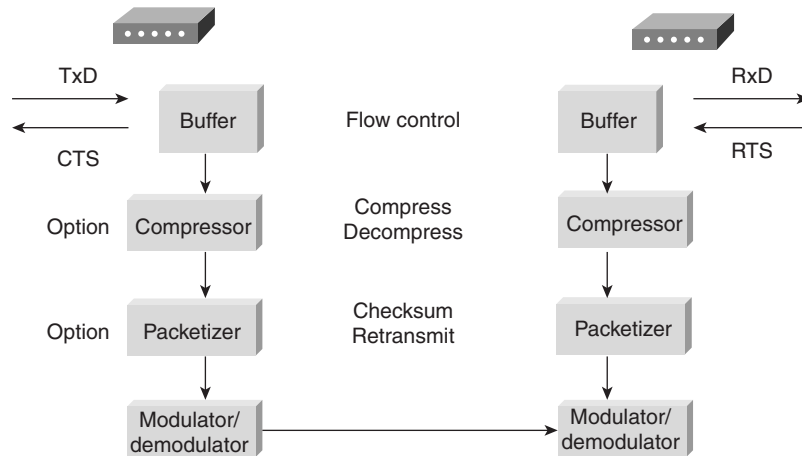
---

- 5 The digital data is modulated into analog signals and sent out through the telephone network.



- 6 When the data reaches the receiving modem, it goes through the same steps in reverse order. The signal will be demodulated; and the data will be depacketized, decompressed, and delivered to the destination DTE. The DTE can use RTS to indicate that it is unable to receive data on RxD.

Figure 4-5 Modem Operation



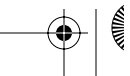
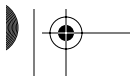
## Communication Wiring and Cabling

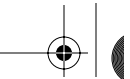
Depending on the type of connections, the cable used can be straight-through, cross-over, or rolled; and it can have a DB-25 or RJ-45 connector.

### DTE-to-DTE Wiring—Null Modem

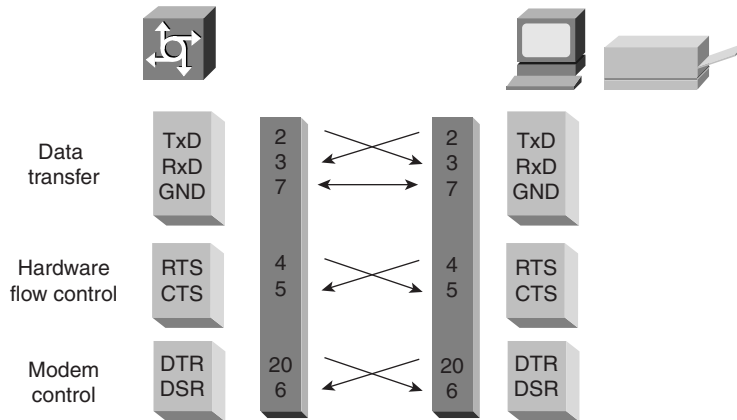
When two DTE devices (for example, an access server and a terminal) are near each other, it makes sense to connect them directly without going through a telephone network and two modems. An ordinary EIA/TIA-232 cable does not work in this case because both DTE devices transmit on the TxD lead (pin 2), and both expect input on the RxD lead (pin 3). A *null modem cable* is required for the DTE-to-DTE connection.

Null modems crisscross DB-25 pins 2, 3, and other corresponding pins so that the two DTE devices can communicate, as shown in Figure 4-6. Some devices can be configured to operate either like a DTE or a DCE. Configuring a device as a DCE usually means that it receives data on pin 2 and transmits data on pin 3.





**Figure 4-6** *Null Modem Cable Pinout*



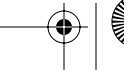
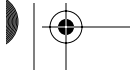
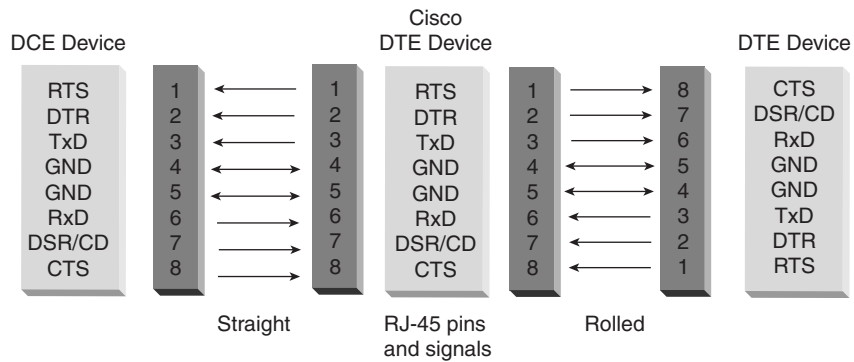
For example, many serial printers are configured as DCE devices so they can be connected directly to a DTE (for example, a PC or a terminal server) with an ordinary EIA/TIA-232 cable, eliminating the need for a null modem connection.

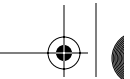
**NOTE** On null modem cable, as seen in Figure 4-6, the CD signal is on the same pin as DSR.

### RJ-45 Wiring

Cisco uses RJ-45 ports and connectors for console, auxiliary, and asynchronous port connections. The specific pinout to be used on a RJ-45 interface for EIA-232 is not defined by any standards. As such, Cisco defines the RJ-45 as a DTE pinout, as shown in Figure 4-7.

**Figure 4-7** *RJ-45 Pinout*





64 Chapter 4: Configuring Asynchronous Connections with Modems

Cabling from the access server port (RJ-45) to an external device (such as a modem or terminal) requires the use of two cabling components:

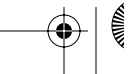
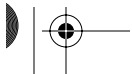
- **RJ-45 to RJ-45 cable**—This can be either a rolled cable (reverse pins 1–8, 2–7, 3–6, 4–5) or a straight-through cable (1–1, 2–2, and so forth). To check whether a cable is straight or rolled, hold the two connectors (the two ends of the cable) side by side—with the keys at the back and the pins up—and compare them by inspecting the color-coded wires inside the connector. If the wires use the same colors on the same pins, it is a straight cable; if the wires are a mirror image of each other, it is a rolled cable. The octal cable used to connect to the asynchronous ports performs the equivalent of a rolled cable.
- **RJ-45 to DB-25 adapter**—This can also be straight-through or rolled. A male or female DTE adapter (MDTE or FDTE) is straight. A male or female DCE adapter (MDCE or FDCE) is rolled. An MMOD adapter is rolled. Only an MMOD adapter supports modems (modified from MDCE connectors by wiring DB-25 pin 8 to DSR, instead of pin 6).

**Difference between Console and Auxiliary Ports**

The console port does not entertain pin 1 (RTS) and pin 8 (CTS), as shown in the following Console Port table. Therefore, hardware flow control is nonexistent. By limiting the speed to 9600 baud, the risk of overflowing the buffer is quasi null. The Auxiliary port uses pins 1 and 8, as shown in the following table, and therefore can accommodate a modem. The source of this information is the Cisco Web site ([www.cisco.com](http://www.cisco.com)). In the following tables, any pin that is not referenced is not connected.

Console Port (DTE)		
Pin	Signal	Input/Output
1	-	-
2	DTR	Output
3	TxD	Output
4	GND	-
5	GND	-
6	RxD	Input
7	DSR	Input
8	-	-

Auxiliary Port (DTE)		
Pin	Signal	Input/Output
1	RTS	Output
2	DTR	Output
3	TxD	Output
4	GND	-
5	GND	-
6	RxD	Input
7	DSR	Input
8	CTS	Input



## Cisco-Specific Connections

The auxiliary and console ports are configured as DTE devices on Cisco access servers. Terminals are also DTE devices. As noted earlier, two DTE devices cannot be directly connected unless the signals are rolled exactly once. So, you must either roll the pins in the cable or in the DB-25 adapters, but not both. The formula for a successful DTE to DTE connection is one of the following:

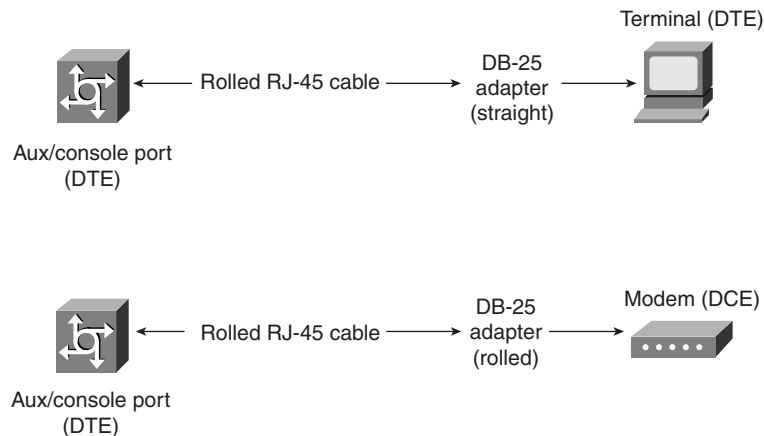
- DTE to a rolled RJ-45 cable to a straight DB-25 adapter to DTE
- DTE to a straight RJ-45 cable to a rolled DB-25 adapter to DTE

When connecting a DTE to a DCE, however, you should have either no rolls, or two rolls in the cable and the connector. The formula for a successful DTE to DCE connection is one of the following:

- DTE to a rolled RJ-45 cable to a rolled DB-25 adapter to DCE
- DTE to a straight RJ-45 cable to a straight DB-25 adapter to DCE

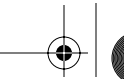
Figure 4-8 shows some possible working connections between an access server and various types of end devices.

**Figure 4-8** *Cisco Working Connections*



The part number for the rolled RJ-45 to RJ-45 cable is CAB-500RJ.

When you order access servers with async ports, you need to order the corresponding cable accessories. Order one CAB-OCTAL-KIT (eight-lead octal cable and eight male DB-25 modem connectors) for each 68-pin async connector that you have on the access server. If the modem uses an RJ-45 connector, you can order one CAB-OCTAL-ASYNC (a rolled eight-lead octal cable with RJ-45 connectors); special adapters might be required.



---

**NOTE** Connecting a modem to the console port of a router is a security risk because it initially has no protection or security features enabled.

---

Cisco routers ship with a console and auxiliary port cabling kit that includes the following components:

- RJ-45 to RJ-45 rolled cable
- RJ-45 to DB-9 female DTE adapter (labeled TERMINAL)—typically used to connect to a PC being used as a console terminal
- RJ-45 to DB-25 female DTE adapter (labeled TERMINAL)—used to connect a terminal to the console or auxiliary port
- RJ-45 to DB-25 male DCE adapter (labeled MODEM)—used to connect the auxiliary port to a modem

Table 4-4 presents the port types for console and auxiliary ports on Cisco routers and switches.

**Table 4-4** *Console and Auxiliary Port Types*

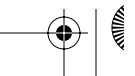
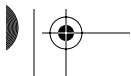
Router	Console Port	Auxiliary Port
DB-25	DCE	DTE
RJ-45	DTE (DCE on the 1700 series)	DTE

## Modem Modulation Standards

The function of a modem is to convert digital signals (DTE-to-DCE) into analog signals (DCE-to-DCE), and vice versa. The ITU-T has defined and introduced several modem modulation standards over the years. Various modem manufacturers, however, have also marketed their own proprietary versions of modems. Interoperability among various types of modems, and sometimes even from the same vendor's modems, can be a challenge.

With proper configuration, V.34 modems can intelligently adapt to line conditions during a transition. Two communicating modems will initially attempt to set up a call at 28.8 Kbps. If line conditions do not allow a transmission at this speed, the modems fall back to the next highest speed in steps of 2.4 Kbps (possibly down to 2.4 Kbps, if necessary). Alternatively, if line conditions improve, the modems can increase the speed.

Older modems negotiate a fixed transmission rate during handshaking, but communications continue at the same speed after that. If line quality deteriorates below a certain threshold, the connection is lost. Older modems cannot take advantage of any increased bandwidth later, when the line quality improves. The V.32bis standard was finalized in July 1991; V.34 standards were finished in June 1994.



The V.34 annex 12 standard supports a 33.6 Kbps transmit and receive operation. When compression is used, up to 133.8 Kbps is possible if the PC can handle this speed.

The newest standard is the V.90, 56 Kbps standard. Most modem manufacturers have a V.90 product, even though the actual maximum data rate allowed is 53 Kbps within the United States.

The access server is unaware of modulations because it is directly involved with DTE-to-DCE communication only. However, the access server-to-modem speed must account for modulation speed and compression ratio for optimal end-to-end performance.

The following are some common international and proprietary modem standards:

*ITU Standards*

V.22: 1200 Bps  
V.22 bis: 2400 Bps  
V.32: 9600 Bps  
V.32 bis: 14.4 Kbps  
V.34: 28.8 Kbps  
V.34 annex 1201H: 33.6 Kbps  
V.90: 56 Kbps

*Proprietary Methods*

V.32 terbo: 19.2 Kbps  
V.fast: 28.8 Kbps  
V.FC: 28.8 Kbps  
K56Flex: 56 Kbps  
X2: 56 Kbps

## Error Control and Data Compression

Error-detection and error-correction methods were developed to ensure data integrity at any speed. Some widely used methods include Microcom Networking Protocol (MNP) and Link Access Procedure for Modems (LAPM).

Compression algorithms typically require error-correction algorithms, so V.42bis and MNP 5 compression usually run over LAPM or MNP 4 correction. V.42 and V.42bis are not limited to V.32 and V.34 modems. They can also be implemented in lower speed equipment. The 4:1 compression ratio provided by V.42bis is theoretical and is rarely achieved.

The modern data-compression technique is analogous to the video-compression or disk-packing algorithms used in computers. The compression efficiency is highly dependent on data content. Some data (such as ASCII files) compress readily, whereas others can be compressed only a little.

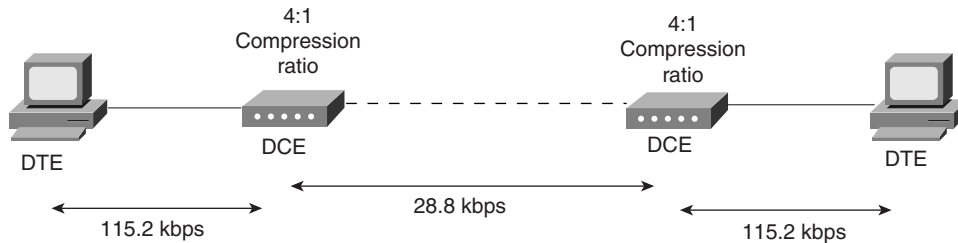
Although some applications' software supports data compression, it is usually better to let the modem compress transmitted data. Data-compression algorithms that run in modem hardware are faster than those performed by host software. If two modems have agreed on V.42bis compression, you need to turn off your application's compression capability. This means transferring data at a higher speed on the interface between the DTE and the DCE.

## 68 Chapter 4: Configuring Asynchronous Connections with Modems

Confusion often arises between the DCE-to-DCE modulation speed and DTE-to-DCE speed. The former represents how fast the modems communicate with each other across the telephone network. The latter represents how fast your computer communicates with the attached modem.

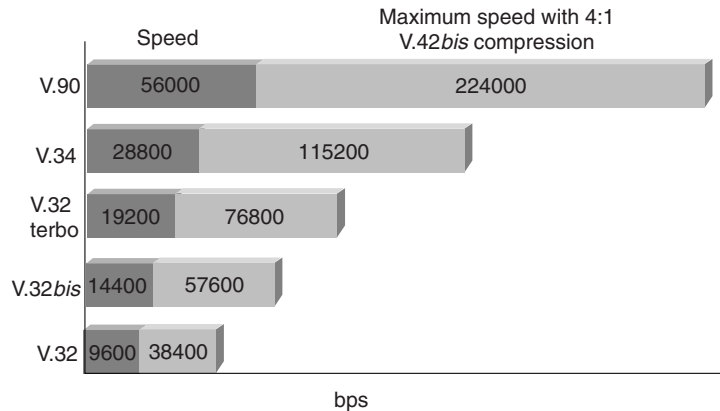
To gain full benefits from compression in an ideal situation, the DTE (for example, a PC) must send to the DCE (for example, a modem) at speeds matching the potential compression ratio (as shown in Figure 4-9). However, the EIA/TIA-232 serial interface (COM port) that is widely found on PCs and some Macintosh computers might operate considerably slower than the full potential of V.34. The problem is that some PCs and Macs use the EIA/TIA-232 serial interface with a combination of Universal Asynchronous Receiver/Transmitters (UARTs) and character-oriented communications software packages, which are not reliable at higher data rates. In a PC, DTE should be set to clock the modem at its fastest rate to take advantage of compression.

**Figure 4-9** *The Speed and Compression Shown Here and Advertised By Modem Manufacturers, Assuming Ideal Conditions*



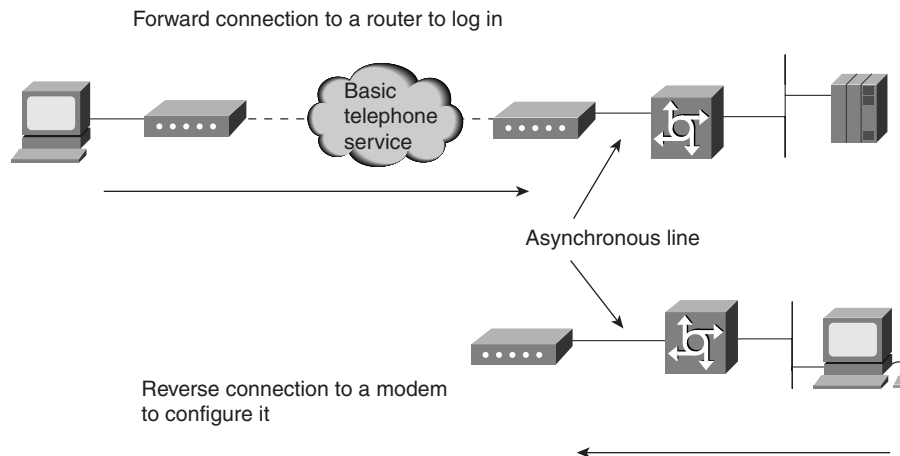
An improperly configured modem might automatically adjust DTE-DCE speeds to match the established DCE-DCE speeds. This is often called *speed mismatch*. To avoid speed mismatch, you must lock the DTE-DCE speed so it remains constant, as originally configured. This speed-locking mechanism is called *speed conversion* (also known as *port-rate adjustment* or *buffered mode*).

Figure 4-10 shows the maximum theoretical speeds that are possible for selected modem-modulation standards. Also shown are the possible speeds if V.42bis compression is used with the same standards.

**Figure 4-10** Theoretical Speeds for Selected Modem-Modulation Standards

## Configuration for Asynchronous Connections

Cisco access servers support both incoming asynchronous line connections (forward connections) and outgoing asynchronous line connections (reverse connections), as shown in Figure 4-11. For example, a remote terminal user that dials into the access server through an asynchronous line makes a forward connection; a user connects through an access server (reverse connection) to an attached modem to configure the modem.

**Figure 4-11** Forward Connection and Reverse Telnet

A host can make reverse Telnet connections to various types of devices attached to a Cisco access server. Different port numbers (20xx, 40xx, and 60xx) are used because different data

70 Chapter 4: Configuring Asynchronous Connections with Modems

type and protocol negotiations will take place for different types of devices attached to the access server.

The remote host must specify a particular TCP port on the router to connect with individual lines or to a rotary group. For example, if you wish to configure a modem connected to the interface Async 7, you will make a reverse Telnet connection using port address 2007. Note that TCP port number 2007 specifies a Telnet protocol connection (TCP port 2000) to line 7. The individual line number is added to the end of the port number type.

**Asynchronous Interfaces—Line Numbering**

Refer to your Router’s manual to see how the lines are counted on that specific platform. As an example, the Cisco 3640 that is represented in the following table has four slots and each of those has preassigned line numbers, as follows:

<b>97-128</b>	<b>65-96</b>
<b>33-64</b>	<b>1-32</b>

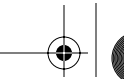
The interface number of a port in a Cisco 3600 is determined by using the following formula:  
 Interface number = (32 × slot number) + unit number + 1

For example, asynchronous port 12 in slot 1 corresponds to interface number 45 ((32 × 1) + 12 + 1 = 45). This is also the line number for the port. Port 12 in slot 1 is always assigned interface number 45, regardless of whether the module in slot 0 is a 16-port asynchronous module, a 32-port asynchronous module, or some other type of module entirely; or even whether there is a network module in slot 0 at all. If you move the module in slot 1 to a different slot, however, its interface numbers change.

Table 4-5 shows the services provided, and the TCP port numbers for individual lines and rotary groups.

**Table 4-5** *Services and Port Numbers for Lines and Rotary Groups*

<b>Services Provided</b>	<b>Base TCP Port for Individual Lines</b>	<b>Base TCP Port for Rotary Groups</b>
Telnet protocol	2000	3000
Raw TCP protocol (no Telnet)	4000	5000
Telnet protocol, binary mode	6000	7000
Xremote protocol	9000	10000



Use the **transport input protocol** command to specify which protocol to use when connecting to a line using reverse Telnet. For example, **transport input all** allows all of the following protocols to be used for the connection: lat, mop, nasi, none, pad, rlogin, Telnet, and v120. Each of these protocols can be specified individually as a command option.

---

### Reverse Telnet—Minimum Configuration

To successfully reverse Telnet to the modem attached to your router, the line interface must have been configured with the **transport input all** and **modem inout** commands.

---

## EXEC Connection Commands

Use the EXEC commands in this section to initiate and control a reverse Telnet terminal session to a modem.

To make a connection with Telnet protocol:

```
Router>Telnet host [port] [/debug]
```

To disconnect the specified session or all sessions:

```
Router>disconnect [session-number]
```

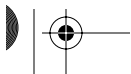
To suspend a session:

```
Router>ctrl-shift-6 x
```

---

EXEC Command	Description
<b>Telnet</b> host [port] [/debug]	Makes a Telnet connection to a host (and optionally to a certain port). The target host can be specified, either by a host name or an IP address. The optional debug switch provides useful information about the connection by displaying the informational level of logging messages. Additionally, you can simply type the name of the host to which you wish to make the connection; by default, an attempt to establish a Telnet session is started. The interface through which the connection is made provides the source IP address for that connection.
<b>disconnect</b> [session-number]	Disconnects the specified connection, or disconnects the most recent connection if not specified.
<b>ctrl-shift-6 x</b>	To suspend the current session, simultaneously press the Ctrl, Shift, and 6 keys, followed by the x key.

---



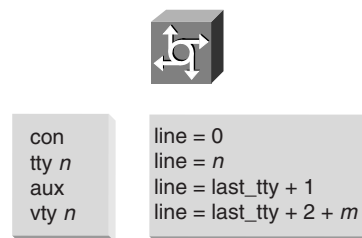
## Line Types and Numbering

Cisco devices have the following four types of lines:

- **CON: Console line**—Typically used to log in to the router for configuration purposes. This line is also referred to as CTY.
- **AUX: Auxiliary line**—RS-232 DTE port used as a backup asynchronous port (TTY). Cannot be used as a second console port.
- **TTY: Asynchronous line**—Same as asynchronous interface. Available on access server models only (Cisco 2509, 10, 11, 12, AS5100, and Cisco 1001). Used typically for remote-node, dial-in sessions that use such protocols as SLIP, PPP, and XRemote.
- **VTY: Virtual terminal line**—Used for incoming Telnet, LAT, X.25 PAD, and protocol-translation connections into synchronous ports (such as Ethernet and serial interfaces) on the router.

Different routers have different numbers of these line types. Figure 4-12 shows the Cisco line-numbering rules, where  $n$  represents the first physical line after the Console line, and  $m$  refers to the number of the vty line. For example, the vty 4 line corresponds to line 14 on a router with eight TTY ports. Because line 0 is for the Console, lines 1 to 8 are the TTY lines, line 9 is for the Auxiliary port, and lines 10 to 14 are for VTY 0 to 4.

**Figure 4-12** Cisco Line Numbering



TTY lines correspond to asynchronous interfaces on a one-to-one basis, and vty lines are virtual lines that are dynamically assigned to the synchronous interfaces. Usually, you would associate vty lines with incoming Telnet sessions.

---

**NOTE** Enter the **interface line tty ?** command to view the maximum number of TTY lines supported.

---

Connections to an individual line are most useful when a dial-out modem, parallel printer, or serial printer is attached to that access server line. To connect to an individual line, the remote

host or terminal must specify a particular Transmission Control Protocol (TCP) port on the access server. If the Telnet protocol is used, that port is 2000 plus the line number. For example:

```
Router#Telnet 131.108.30.40 2001
```

This command indicates a Telnet connection to line 1 (2000 + 1).

---

### Line Numbering On Cisco 1600

Some routers don't have AUX ports, and the Cisco 1600 is one of them. The following shows the way the relative and absolute line numbers are presented with the **show line** command:

```
Router#show line
Tty Typ Tx/Rx A Modem Roty Acc0 AccI Uses Noise Overruns
* 0 CTY - - - - - 0 1 0/0
  2 VTY - - - - - 0 0 0/0
  3 VTY - - - - - 0 0 0/0
  4 VTY - - - - - 0 0 0/0
  5 VTY - - - - - 0 0 0/0
  6 VTY - - - - - 0 0 0/0
Line(s) not in async mode -or- with no hardware support: 1
```

The CTY port is the Console. As shown in Figure 4-12, the AUX port receives the number TTY + 1. Because this Cisco 1600 router has no Async interface (no TTY), the AUX port, if present, would have received 1 line number 1. The VTY lines are always as follows: Last\_TTY + 2. Using the formula shown on Figure 4-12 to find the first VTY line number, calculate 0 TTY + 2 = 2, which is the starting number of VTY lines. By default, the router provides five virtual connections; in this case, these are numbered 2, 3, 4, 5, and 6.

You can use the **show line** command to display all types of lines and the status of each line, as exhibited in Figure 4-13. It also provides useful information about modem control and asynchronous port configuration. The **show line line-number** command displays more detailed information on the specified line, which includes some useful data such as baud rate, modem state (idle or ready), and modem hardware state (CTS, DSR, DTR, and RTS for hardware flow control and session control). Table 4-6 explains the output fields displayed in Figure 4-13. Figure 4-13 emphasizes concepts previously discussed in this book, with the exception of Access Class.

---

### Filtering Traffic on VTY Lines—Access Class

If you wish to restrict incoming and outgoing connections between a particular virtual terminal line (into a Cisco device), you can use the **access-class** command on a line. The **access-class** command makes a standard access-list decide whether it should accept or reject a connection.

74 Chapter 4: Configuring Asynchronous Connections with Modems

Remember to set identical restrictions on all the virtual terminal lines because a user can connect to any of them.

Figure 4-13 Sample Output of the show line Command

Rotary Group #	Tty	Type	Tx/Rx	[A] Modem	Roty	Acc0	Accl	Uses	Noise	Overruns	Access class in/out
	* 0	CTY		-	-	-	-	0	0	0/0	
	* 1	TTY	115200/115200	- inout	-	4	-	31	26	0/0	
	* 2	TTY	115200/115200	- inout	-	30	-	37	23	0/0	
Autoselect state	A 3	TTY	115200/115200	- inout	-	25	-	10	24	1/0	
	* 4	TTY	115200/115200	- inout	-	4	-	20	63	1/0	
	* 5	TTY	115200/115200	- inout	-	45	-	18	325	22/0	
	A 6	TTY	115200/115200	- inout	-	25	-	7	0	0/0	
	7	TTY	115200/115200	- inout	-	-	-	6	36	1/0	
	8	TTY	115200/115200	- inout	-	-	-	3	25	3/0	Modem setting
	* 9	TTY	115200/115200	- inout	-	4	-	2	0	0/0	
Absolute line number	A 10	TTY	115200/115200	- inout	-	56	-	2	470	216/0	
	11	TTY	115200/115200	- inout	-	-	-	0	0	0/0	
	12	TTY	115200/115200	- inout	-	-	-	0	0	0/0	
	13	TTY	115200/115200	- inout	-	-	-	1	0	0/0	
	14	TTY	115200/115200	- inout	-	-	-	0	0	0/0	
	15	TTY	115200/115200	- inout	-	-	-	0	0	0/0	
	16	TTY	115200/115200	- inout	-	-	-	0	0	0/0	
	17	AUX	9600/9600	-	-	-	-	2	1	2/104800	
Line speed	* 18	VTY	9600/9600	-	-	-	-	103	0	0/0	
	19	VTY	9600/9600	-	-	-	-	6	0	0/0	
	20	VTY	9600/9600	-	-	-	-	1	0	0/0	
This is vty2 (3rd vty) line 20	21	VTY	9600/9600	-	-	-	-	0	0	0/0	
	22	VTY	9600/9600	-	-	-	-	0	0	0/0	
	23	VTY	9600/9600	-	-	-	-	0	0	0/0	
	24	VTY	9600/9600	-	-	-	-	0	0	0/0	
	25	VTY	9600/9600	-	-	-	-	0	0	0/0	
	26	VTY	9600/9600	-	-	-	-	0	0	0/0	
	27	VTY	9600/9600	-	-	-	-	0	0	0/0	
	28	VTY	9600/9600	-	-	-	-	0	0	0/0	
	29	VTY	9600/9600	-	-	-	-	0	0	0/0	
	30	VTY	9600/9600	-	-	-	-	0	0	0/0	
	31	VTY	9600/9600	-	-	-	-	0	0	0/0	
	32	VTY	9600/9600	-	-	-	-	0	0	0/0	
	33	VTY	9600/9600	-	-	-	-	0	0	0/0	Number of TCP connections made

**Table 4-6** *show line Field Descriptions*

Field	Description
Tty	Line number. An asterisk (*) preceding the number in the Tty field means that the line is currently active, running a terminal-oriented protocol.  An "A" preceding the number indicates that the line is currently active in asynchronous mode.  An "I" preceding the number indicates that the line is configured, but is inactive.
Typ	Type of line. Possible values are CTY-console, AUX-auxiliary port, TTY-asynchronous terminal port, and lpt-parallel printer.
Tx/Rx	Transmit rate/receive rate of the line.
A	Indicates whether autobaud is configured for the line. A value of F indicates that autobaud has been configured; a hyphen indicates that it is not configured.
Modem	Type of modem signal that has been configured for the line. Possible values include callin, callout, cts-req, DTR-Act, inout, and RIisCD.
Roty	Rotary group configured for the line.
AccO, AccI	Output or input access-list number configured for the line.
Uses	Number of connections established to or from the line since the system was restarted.
Noise	Number of times noise has been detected on the line since the system was restarted.
Overruns	Hardware (UART) overruns or software buffer overflows, both defined as the number of overruns or overflows that have occurred on the specified line since the system was restarted. Hardware overruns are buffer overruns; the UART chip has received bits from the software faster than it can process them. A software overflow occurs when the software receives bits from the hardware faster than it can process them.

## Router Configuration

Some configuration is required on a router so it can place and receive asynchronous calls.

### Interface Async and Line Configuration

There is often confusion between the **interface async** and **line** commands. The major difference is that the **interface async** command lets you configure the protocol (logical) aspects of an asynchronous port, as follows:

```
Router(config)#interface async 8
Router(config-if)#encapsulation ppp
```

## 76 Chapter 4: Configuring Asynchronous Connections with Modems

```
Router(config-if)#async dynamic address
Router(config-if)#async mode interactive
Router(config-if)#ppp authentication chap
```

The **line** command lets you configure the physical aspects of the same port, such as the modem attached to it:

```
Router(config)#line 8
Router(config-line)#login local
Router(config-line)#modem inout
Router(config-line)#speed 115200
Router(config-line)#flowcontrol hardware
Router(config-line)#rotary 33
Router(config-line)#autoselect ppp
```

The **async** commands can be thought of as internal configuration; the **line** commands configure external characteristics of the configuration. For example, you configure the basic modem-related parameters on an access server using the **line** command, but you configure the protocol encapsulation and authentication schemes with the interface **async** command.

### Basic Async Configuration—Router Preparation

To make a successful asynchronous connection, you need to configure the modem and the access server properly.

On the access server (router) use the following commands to configure the line to which the modem is attached:

```
(config)#line x
(config-line)#login
(config-line)#password password
(config-line)#flowcontrol hardware
(config-line)#speed 115200
(config-line)#transport input all
(config-line)#stopbits 1
(config-line)#modem inout
```

OR

```
(config-line)#modem dialin
```

Command	Description
<b>line x</b>	Used to identify a specific line for configuration and to start the line configuration command collection mode.
<b>exec</b>	Allows the EXEC process on this line. The EXEC process is activated automatically on all lines. Therefore, because this command is the default, it does not appear in the configuration.
<b>login</b>	Sets a login password on this line. Without the password, no connection is allowed.

Command	Description
<b>password</b> <i>password</i>	Sets the password to be used when logging in to this line.
<b>flowcontrol hardware</b>	Uses RTS/CTS for flow control.
<b>speed</b> <b>115200</b>	Sets the maximum speed (in bits-per-second) between the modem and the access server. The <b>speed</b> command sets both the transmit and receive speed.
<b>transport input all</b>	Allows all protocols to be passed to the access server through this line.
<b>stopbits</b> <b>1</b>	Sets the number of stop bits transmitted per byte.
<b>modem inout</b>	Uses the modem for both incoming and outgoing calls.
<b>modem dialin</b>	For incoming calls only. This is the default.

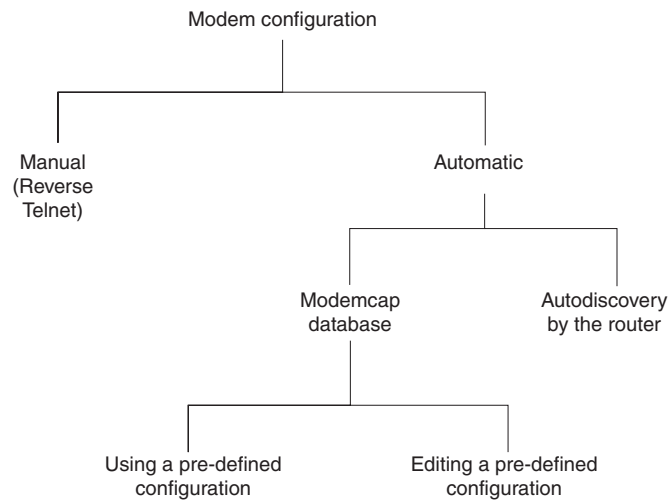
**WARNING** Software flow control (xon and xoff characters) is not recommended with modems and Cisco routers.

## Basic Async Configuration—Modem Preparation

A modem can be configured in many different ways to work with a router.

Figure 4-14 shows that you can manually configure a modem by sending AT commands. You may want to have the router automatically discover and configure the modem, or you may wish to specify to the router which configuration commands to send to the modem.

**Figure 4-14** *Modem Configuration Can Be Done Manually or Automatically*



All these configuration methods are explored in greater detail in the following sections.

## Manual Configuration of Modems

You can elect to manually configure the modem instead of having the router force a configuration on it.

### Standard Commands

On the modem side, you can use the standard configuration commands to do the following:

- Perform hardware flow control.
- Lock DTE speed to ensure that the modem will always communicate with the access server at the specified speed. As an example, when you use an async interface, you lock the speed to its theoretical maximum of 115.2 kbps. The router speed command sets both transmit and receive speeds.
- Hang up when you quit a session.
- Have the Carrier Detect signal truthfully reflect the carrier state.

To manually configure your modem, you most likely reverse Telnet to your modem and apply some AT commands. AT stands for *attention commands* for the modem. In general, each modem vendor has its own modem command set that differs from other vendors' command sets. However, the following modem commands are common among most vendors:

Command	Description
AT&F	Loads the factory default settings (read-only).
ATS0=1	Sets the modem to automatically answer all incoming calls on the first ring. (Recommended to be set to 2 for lines with caller ID.) <sup>1</sup>
AT&C1&D3	Sets up modem control (CD and DTR).
ATS2=255	Ignore the +++ command. The +++ characters set the modem to command mode. You might need to configure the far-end modem to ignore +++ because the +++ command issued to the near-end modem will be transmitted to the far-end modem. The far-end modem might interpret it and cause the connection to hang. This is a bug in the far-end modem. Many modems might operate this way.
ATE0	When <b>echo off</b> is set, the modem will not echo keystrokes.
ATM0	Turns off the external audio output from the modem.

1. Warning: Because the Caller ID function activates on the second ring, hackers typically target modems that answer on the first ring. If a modem takes more than one ring to answer, many hackers don't pursue the matter. You are therefore advised to set your modem to at least ATS0=2, thus pretending that your line subscribes to Caller ID.

---

### AT Command Sets

AT commands stands for attention commands that are used to address the modem and force a certain behavior from it. The following is a listing of different command sets that exist and examples of commands used in the modem modules of Cisco 2600 and Cisco 3600 series:

- **Alphabetic** Command Set—**Q0** enables output of result codes.
- **Caret** Command Set—**^V** Displays the modem's current flash memory bootstrap revision.
- **Dollar** Command Set—**\$Bn** sets the serial port rate to *n* bps, for example **\$B 115200**.
- **Percent** Command Set—**%Bn** sets the modem port data rate, for example **%B 33600**.
- **Ampersand** Command Set—**&D3** sets the modem on-hook and resets when detecting an on-to-off transition of DTR.
- **Parenthesis** Command Set—**)M1** autoadjusts the power level for cellular modems.
- **Asterisk** Command Set—**\* I** displays the modem identifier.
- **Hyphen** Command Set—**-DL** redials the last number dialed.
- **Colon** Command Set—**:Dn** sets manual dial.
- **@** Character Command Set—**@E** gives detailed modem call status.
- **Backslash** Command Set—**\F** displays stored phone numbers.
- **#** Character Command Set—**#E0** enables the escape code sequence in answer mode.

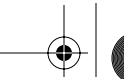
For more info on the AT Commands, consult [http://www.cisco.com/univercd/cc/td/doc/product/access/acs\\_mod/cis3600/analogfw/analogat.htm#12195](http://www.cisco.com/univercd/cc/td/doc/product/access/acs_mod/cis3600/analogfw/analogat.htm#12195).

---

### Nonstandard Modem Commands

Many modem commands are not standardized, and they vary from one vendor to another. The following modem configurations and commands are essential for modems that are attached to Cisco access servers:

- **Hardware flow control**—Use CTS and RTS.
- **Lock DTE speed**—Sets the serial port of the modem to a fixed data transfer rate. Locking the speed between the modem and DTE device prevents the speed from being negotiated down during the initial call setup.
- **Error correction**—Sets error control.
- **Compression**—Uses the best compression algorithm that can be negotiated between the two communicating modems.
- **Show configuration**—Shows current modem settings.
- **Getting help**—Shows all of the AT commands for your specific modem.



80 Chapter 4: Configuring Asynchronous Connections with Modems

- Saving the configuration—Saves the configuration that you just entered in the NVRAM of the modem.

Table 4-7 shows some standard modem initialization strings. For nonstandard modem commands, you must refer to the vendor’s user manual that comes with each modem you purchase. More information about commands for each modem type can be found at <http://www.cisco.com/warp/public/701/21.html>.

**Table 4-7** *Standard Modem Commands*

Command	Microcom	Hayes	USR
Hardware flow control	AT\Q3	AT&K3	AT&H1&R2
Lock DTE speed	AT\J0	AT&Q6	AT&B1
Error correction	AT\N6	AT&Q5	AT&M4
Compression	AT% C1	AT&Q9	AT&K1
Show configuration	AT\S1	AT&V	ATI4
Getting help <sup>1</sup>	AT\$H	AT\$H	AT\$
Saving the configuration	AT&W	AT&W	AT&W

1. The help feature is not always built in. As an example, I could not get the help on a new PCMCIA 10/100 + 56K Modem card I had, no matter what help command I tried. I called the manufacturer, a major PCMCIA cards provider, to find out that indeed no help had been built for the modem portion of its new combo cards.

**Modem Initialization Strings**

Initialization strings are used to send commands to modems before they dial out. They are a series of parameter settings that are sent to your modem to configure it to interact with the access server in a specified way. No strings are required when you dial into a modem. The following are some examples of modem initialization strings:

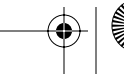
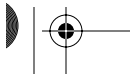
U.S. Robotics (USR) Courier	at&fs0=1&c1&d3&h1&r2&b1&m4&k1&w
Hayes Optima/Accura	at&fs0=1&c1&d3&k3&q9&w
Microcom QX4232 series	at&fs0=1&c1&d3\q3\j0\n6%c1&w

There are numerous other modems from different vendors. Always refer to the user manual from your modem vendor for the proper modem commands to use.

Note that the AppleTalk Remote Access (ARA) 1.0 (or 2.0) client software includes built-in MNP 4 capability. It might have problems if MNP 4 is enabled on the answering modem. Some modems let you disable MNP, but allow LAPM.

For example, you can use the following command strings to disable MNP:

USR Sportster series	ats27=16
Practical Peripherals MiniTower series	at&q9



Also, for a list of AT commands for various modem types and their specific initializing strings, refer to Appendix D, "AT Commands for Modems and Chat-Scripts."

## Automatic Configuration of Modems

Modem autoconfiguration facilitates the configuration of modems on access servers. To set up a modem using modem autoconfiguration, connect the phone line and power cable to the modem, and use the **modem autoconfigure** command on the line with the modem. No other setup function is required for most modems.

To better understand modem autoconfiguration, this section contains the following topics:

- Modem capability database (modemcap file in the Cisco IOS software)—Modemcap is a database of modems and their modem configuration command strings.
- Modem autodiscovery—You can configure a line to automatically attempt to discover the type of modem on the line and to use that modem configuration.
- Modem autoconfiguration—You can configure a line to use a specified modem type.

## Modem Capability Database

The modem capability database (modemcap) is a list of modems with a known set of AT configuration commands for setting each modem type's attributes. For example, many modems use the string **AT&F** to reset the modem to its factory default attributes.

Modem attributes have a full name and a two- or three-letter abbreviation.

Factory default, for example, is also referred to as FD. For normal operation, you need not know these abbreviations. If you are familiar with the modem abbreviations, you can add entries to the modemcap database.

The modemcap database contains entries for supported modems. You can do the following tasks to manage a modemcap database entry:

- View modem entries in the modemcap database with the **show modemcap** command.
- View the contents of a modem's modemcap entry.
- Modify a modem's modemcap entry.
- Create a modem database entry.

The **show modemcap** command displays the modems in the modemcap database, as follows:

```
Router#show modemcap
default
codex_3260
usr_courier
usr_sportster
hayes_optima
```

## 82 Chapter 4: Configuring Asynchronous Connections with Modems

---

```
global_village
viva
telebit_t3000
microcom_hdms
microcom_server
nec_v34
nec_v11
nec_piafs
cisco_v110
mica
```

In addition, with the modem type specified, it shows a complete list of the specified modem's modemcap entry that includes the following fields:

- Command description
- Command abbreviation (with colon separator)
- Command string

The command **show modemcap codex\_3260** shows the AT command string attributes and their values for the Codex 3260 modem, as shown in the following:

```
Router#show modemcap codex_3260
Modemcap values for codex_3260
Factory Defaults (FD): &F
Autoanswer (AA): S0=1
Carrier detect (CD): &C1
Drop with DTR (DTR): &D2
Hardware Flowcontrol (HFL): *FL3
Lock DTE speed (SPD): *SC1
Best Error Control (BER): *SM3
Best Compression (BCP): *DC1
No Error Control (NER): *SM1
No Compression (NCP): *DC0
No Echo (NEC): E0
No Result Codes (NRS): Q1
Software Flowcontrol (SFL): [not set]
Caller ID (CID): &S1
Miscellaneous (MSC): [not set]
Template entry (TPL): default
```

The default modem type has modemcap values for a few of the most common attributes. It does not contain strings for attributes that vary widely by modem type, such as locking speeds, setting hardware flow control, or dealing with compression and error correction.

You can use the **modemcap entry** *modem\_name* command or the **show modemcap** *modem\_name* command to see the contents of a modem's modemcap entry. The **modemcap entry** *modem\_name* command displays modemcap values in a truncated form.

As you will see later in this section, you can also create variant modemcap entries to add new modems or to extend a modem's functionality in the modemcap database.

## Modem Autodiscovery

If no modem is specified for a particular line and you have provided the **modem autoconfigure discovery** command, the access server attempts to autodiscover the type of modem to which it is attached. The access server determines the type of modem by sending AT commands to the modem and evaluating the response. The Cisco IOS software initially tries the first of the modemcap strings to see if the modem initializes properly. If not, the Cisco IOS software cycles to the next string and repeats the process until the appropriate string is found. Usually, if none of the strings properly initializes the modem, you must manually configure the modem.

---

**NOTE** Sometimes, the router fails to recognize a modem, even though it might be part of the modemcap list. Therefore, if you know that your modem can be configured by using an initialization string from one of these scripts, you can issue the **modem autoconfigure type type** command, as explained in the next section.

---

The following is an example of the **modem autoconfigure discovery** command:

```
Router(config)#line 1 16
Router(config-line)#modem autoconfigure discovery
```

This command instructs the access server to do the following on lines 1 through 16:

- Send the AT string at various baud rates until it receives an OK.
- Send a variety of AT commands that attempt to receive a complete identification of the modem identified in the access server's modem capabilities database.

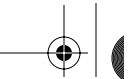
The specific modemcap entries found on a particular system are determined by the hardware and Cisco IOS software version that are installed.

---

**NOTE** Whenever possible, configure the modem to eliminate the overhead of modem autodiscovery. If you list a specific modem type, initialization proceeds more quickly.

---

If the access server cannot determine the modem type, the default modem entry is used. Any modems that are not currently supported in the list can be manually added to the list to be autodiscovered in future communication, as you will see later in the section, "Fine-tuning Modem Autoconfiguration."



### Modem Autoconfiguration

If you know that your modem can be configured by using an initialization string from one of the preconfigured scripts of the modemcap, you can issue the **modem autoconfigure type type** command. In this command, *type* is one of the strings in the modemcap list displayed in the "Modem Capability Database" section, earlier in this chapter. If you list a specific modem type, initialization proceeds more quickly than if you let the autodiscovery take place. In the following example, the access server is configured to send an initialization string for a USR Sportster modem on line 1:

```
Router(config)#line 1
Router(config-line)#modem autoconfigure type usr_sportster
```

---

#### NOTE

To eliminate the overhead of modem autodiscovery and to avoid modem configuration ambiguity caused by modem autodiscovery, configure the modem type by using the **modem autoconfigure type** command whenever possible.

---

With automatic modem configuration, each time a modem is reset, a chat-script is executed that sends a string of modem configuration commands (AT commands) to the modem. This modem configuration command string is generated automatically whenever the modem is recycled.

---

#### Speed Issues with Modem Autoconfiguration

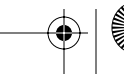
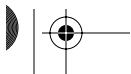
Here are the results of autoconfiguration tests that were performed on 3640s to determine how long it took a Cisco Router to configure an attached modem:

Test Description	Time
Autodiscovery with no match found (therefore, default settings were applied), unknown discovery	6 seconds
Autodiscovery with a match found (USR Sportster), known discovery	5 seconds
Autoconfigure with modem type specified (type USR)	2 seconds

---

### Fine-tuning Modem Autoconfiguration

If none of the strings from the modemcap properly initializes the modem, you must manually configure the modem or change the modemcap database.

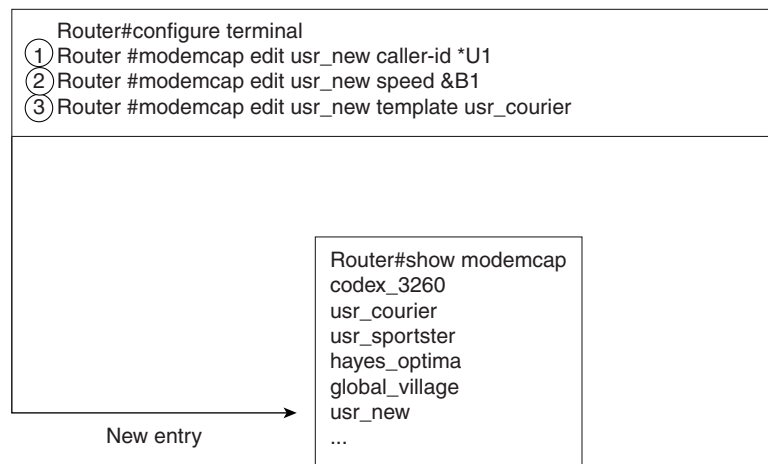


Use the **modemcap edit** *new\_modem\_name* command to do the following:

- Add a new entry to the modemcap database
- Add new attributes to an existing modem entry in the modemcap database

Figure 4-15 shows the uses of the **modemcap edit** *usr\_new* command. The command line marked with a **1** creates the *usr\_new* entry in the modemcap database and sets the caller-id for the *usr\_new* modem to \*U1. The command line marked with a **2** locks the DTE speed on this modem. Finally, the command line marked with a **3** points to another modemcap entry to be used as a template. As a result, any value not found in the current modemcap entry is set by the template modemcap entry. In this example, the **usr\_courier** modemcap entry is the template. You can have up to four layers of templates.

**Figure 4-15** *Creating and Editing a Variant Modemcap Entry*



Additional commands that can be used when creating variant modem cap entries are as follows:

- The **modemcap edit** command edits user-created modemcap entries only.
- Use the **no modemcap edit** *modemname* command to remove the specified modem from the modemcap database. With this command, the whole modemcap entry is deleted.
- The **no modemcap edit** *modemname attribute* command removes a modem attribute from a modem's modemcap entry without deleting the entry altogether.

After configuring a modemcap entry with the **modemcap edit** command, use the **show modemcap** *modem\_name* command to verify the new modemcap attribute values.

Figure 4-16 shows the output for the new modemcap created in Figure 4-15.

**Figure 4-16** Viewing a Variant Modemcap Entry

```

Router#show modemcap usr_new
Modemcap values for usr_new
Factory Defaults (FD) : &F
Autoanswer (AA) : S0=1
Carrier detect (CD) : &C1
Drop with DTR (DTR) : &D2
Hardware Flowcontrol (HFL) : &H1&R2
② Lock DTE speed (SPD) : &B1
Best Error Control (BER) : &M4
Best Compression (BCP) : &K1
No Error Control (NER) : &M0
No Compression (NCP) : &K0
No Echo (NEC) : E0
No Result Codes (NRS) : Q1
Software Flowcontrol (SFL) : [not set]
① Caller ID (CID) : *U1
Miscellaneous (MSC) : [not set]
③ Template entry (TPL) : usr_courier

```

The numbers in the Figure 4-16 output correspond to the numbers in Figure 4-15 that show each **modemcap edit** command from the example.

Specifically, the `usr_new` modemcap shown in Figure 4-16 is identical to the `usr_courier` entry, with the following exceptions:

- The DTE speed lock
- The caller ID field
- The template

If you use the **show running-config** command, the `usr_new` information for the configuration on the previous page appears as follows:

```
usr_new SPD=&B1:CID=*U1:TPL=usr_courier
```

## Chat-Scripts for Async Lines

Because asynchronous modems are not standard, you must write custom chat-scripts to perform certain tasks. Chat-scripts are used for the following tasks:

- Modem configuration
- Dialing and remote login commands
- Failure detection

A **chat-script** is a string of text that defines the handshaking that occurs between two DTE devices, or between a DTE and its directly attached. It consists of expect-send pairs that define

the string that the local system expects to see from the remote device and which reply the local system should send:

```
(config)#chat-script script-name expect-string send-string
```

For example, you can configure chat-scripts for the following tasks:

- Initializing the directly attached modem
- Instructing the modem to dial out
- Logging in to a remote system

The following is a sample **chat-script** command and a table that describes it:

```
(config)#chat-script Reno ABORT ERROR ABORT BUSY "" "ATZ" OK "ATDT \T" TIMEOUT 30
↳CONNECT \c
```

<b>chat-script Command</b>	<b>Description</b>
<b>Reno</b>	Defines the name of this chat-script as dial.
<b>ABORT ERROR</b>	Stops the chat-script if an error is encountered.
<b>"ATZ"</b>	Without expecting an input string, sends the AT command to the modem to reset it by using the stored profile.
<b>OK "ATDT \T"</b>	When the input string OK is seen, sends the AT command to instruct the modem to dial the telephone number in the dialer string or <b>start-chat</b> command.
<b>TIMEOUT 30 CONNECT</b>	Waits up to 30 seconds for the input string CONNECT.
<b>\c</b>	Suppresses a new line at the end of the send string. It is an escape sequence. Indicates the end of the chat-script.

For more information on chat-scripts, please refer to Appendix D, "AT Commands for Modems and Chat-Scripts."

### Modem-script versus System-script

Chat-scripts are used as modem-scripts or system-scripts. Modem-scripts are used between DTE to DCE, where system-scripts are sent DTE to DTE.

In the following example, the script called Niagara is used between the router and the modem to successfully handshake with the destination. The Gambling script is used for logging between the router and an end-system at the destination; ! script called Niagara is used for dialing a modem:

```
chat-script Niagara ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 30 CONNECT \c
!
! Script for logging into a system called Gambling and starting up slip session:
```

## 88 Chapter 4: Configuring Asynchronous Connections with Modems

```
chat-script Gambling ABORT invalid TIMEOUT 15 name: billw word: wewpass ">" "slip
↳default"
!
Interface async 5
dialer map ip 172.16.12.17 modem-script Niagara system-script Gambling
↳918005551212
```

You can manually start a chat-script on any asynchronous line that is not currently active by using the **start-chat** command. Or, you can configure chat-scripts so they are executed for specific events, such as the following:

- Line activation—Triggered by incoming traffic (Carrier Detect signal going up)
- Connection—Triggered by outgoing traffic (for example, reverse Telnet)
- Line reset—Triggered by async line reset
- Startup—Triggered by access server startup
- Dialer—Triggered by dial-on-demand routing (DDR)

### Start-chat: Manual Start of Chat-script

If you wish to manually start a chat-script on a line, you can use the **start-chat** privileged EXEC command.

```
Router#start-chat regexp [line-number [dialer-string]]
```

This command provides modem dialing commands for a chat-script that you want to apply immediately to a line. If you do not specify a line, the script runs on the current line. If the specified line is already in use, the script is not activated, and an error message appears.

The argument *regexp* is used to specify the name of the modem script that is to be executed. The first script that matches the argument in this command and the **dialer map** command will be used.

## Verifying and Debugging Modem Autoconfiguration

The **debug confmodem** command displays the modem configuration process, as shown in Example 4-1. Example 4-1 shows an access server modem configuration process on line 97 with a USR Sportster modem attached.

### Example 4-1 Modem Configuration Process—**debug confmodem**

```
Router#debug confmodem
TTY97: detection speed (115200) response ---OK---
TTY97: Modem command: --AT--
TTY97: Modem configuration succeeded
TTY97: Detected modem speed 115200
```

**Example 4-1** *Modem Configuration Process—debug confmodem (Continued)*

```
TTY97: Done with modem configuration
TTY97: detection speed (115200) response ---OK---
TTY97: Modem command: --AT&F&C1&D2&H1&R2&M4&K1&B1S0=1H0--
TTY97: Modem configuration succeeded
TTY97: detection speed (115200) response ---OK---
TTY97: Done with modem configuration
```

You can also use the following commands to verify operations:

- The **show line** command shows the type of modem configured on a line.
- The **clear line** command returns a line to its idle state. Normally, this command returns the line to its conventional function as a terminal line with the interface left in a down state.

## Troubleshooting Modem Autoconfiguration

To troubleshoot modem autoconfiguration, consider the following conditions and solutions: the modem is not responding, the modem is not recognized by modem autoconfigure discovery, or there is an original modemcap entry problem.

If the modem is not responding:

- Is the modem plugged in and turned on?
- Is the power-up configuration set to factory default?
- Can you connect the modem through reverse Telnet?
- Do you have a dial tone at the phone jack?

If the modem is not recognized by modem autoconfigure discovery:

- Use the **show line** command to verify the modem configuration that the line is using.
- Does the Cisco access server recognize the modem?
- Use the **modem autoconfigure type** *modem-name* command.

**NOTE**

Use the **show modemcap** command to verify modemcap support for this modem.

If there is an original modemcap entry problem:

- If you configured your own modemcap entry and reconfiguration appears to function, verify that the DTR attribute is not set to &D3.

And as a last resort, do not forget that you can also check the modem manufacturer's manual.

## Summary

You learned in this chapter how to configure an access server for modem connectivity. To do this, you saw how to perform a reverse Telnet session to the modem, how to configure it for basic asynchronous operations, and how to use the modem autoconfiguration.

In Chapter 5, "Configuring Point-to-Point Protocol and Controlling Network Access," you will learn how to establish a PPP session over the lines you have configured for Async connectivity.

## Case Study 4-1—Configuring Asynchronous Connections with Modems

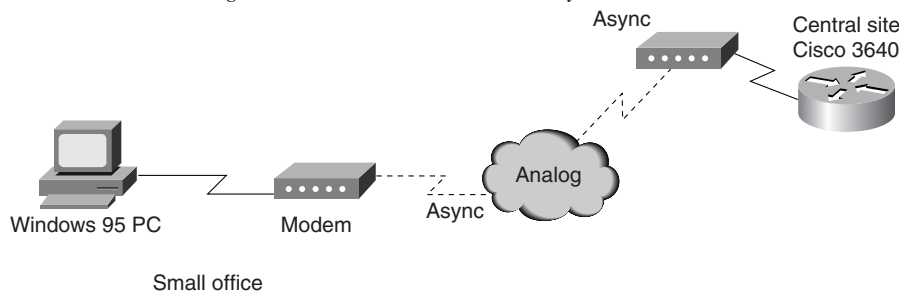
In this case study, you have to set up an initial configuration for an access server to provide modem support. You then manually configure the attached modem. Finally, you are required to verify proper operation by dialing the connection from a PC. From the Central site's router, you verify the connected users and list the status of the modem hardware control signals.

### Scenario

Complete the tasks of this case study, and then review the case study solution section that follows to see how you did and see where you might need to review the concepts presented in this chapter.

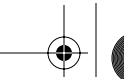
You were asked to configure the external modem on the central site router (Cisco 3640) to receive an asynchronous call placed from the PC, as shown in Figure 4-17.

**Figure 4-17** *Telecommuter Calling Back at the Central Site via an Asynchronous Connection*



### Task 1—Setting Up the Initial Configuration of the Central Site Router

Configure this new router for the first time. Provide it with all parameters necessary during an initial configuration. For example, provide the hostname, ip address, routing protocol, and so on.



## Task 2—Configuring the Serial Interface and Line

**Step 1** Configure the serial interface to which the external modem is connected, and the corresponding line.

**Step 2** From the global configuration mode, configure a hostname to allow reverse.

## Task 3—Configuring the Central Site Modem

**Step 1** Enter **modem** at the command prompt (Reverse Telnet to line 97 of the Cisco 3640).

**Step 2** Enter **AT** and press Return. Observe that you receive an OK from the modem.

**Step 3** Enter and describe what each of the following AT commands do (you can use the modem built-in help, the modem manual, or visit the Web site of the modem manufacturer to find the answers):

<b>AT\$</b>	<b>AT&amp;\$</b>	<b>AT\$\$</b>
<b>AT&amp;F1</b>	<b>ATS0=1</b>	<b>AT&amp;C1</b>
<b>AT&amp;D2</b>	<b>AT&amp;H1</b>	<b>AT&amp;R2</b>
<b>AT&amp;M4</b>	<b>AT&amp;B1</b>	<b>AT&amp;K1</b>
<b>AT&amp;W0</b>	<b>ATI4</b>	

## Task 4—Configuring the PC Modem and Connecting to Central Site

From the remote PC, initiate a reverse Telnet session on the PC modem, and dial in to the central router. Suppose that the phone number to reach CentralC is a four-digit speed-dial number: 1005.

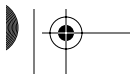
Once the modems have synchronized (trained), you should be prompted with a user access verification in the pmodem window. Log in, and go into the privileged EXEC mode. Verify users' connectivity to port 97. Also, verify that the Central site router is properly configured.

## Task 5—Simplifying Router Modem Configuration with Autoconfigure

**Step 1** Turn on a debugging function so you can see the router autoconfiguring the modem.

**Step 2** Configure your line 97 so the access server will attempt to autoconfigure the modem as a US Robotics Sportster.

**Step 3** Save your configuration.



## Solution to Case Study 4-1—Configuring Asynchronous Connections with Modems

The following is a step-by-step discussion of the case study solution.

### Task 1 Solution—Setting Up the Initial Configuration of the Central Site Router

Complete the following steps on the Central site router.

#### NOTE

Some commands that are requested in this case study do not pertain to Async connectivity; they pertain to the initial configuration of any Cisco routers. This is necessary because your central site access server was never configured prior to this case study.

Commands requested during case studies that do not relate to previous chapters are usually commands covered in ICRC and ACRC courses and books.

Configure your Central site router for the following initial configuration:

Command	Description
<b>hostname</b> <i>hostname</i>	Provides your router with a name. For this case study, use CentralA.
<b>no ip domain-lookup</b>	Avoids domain lookups of typos.
<b>router igrp</b> <i>100</i>	Enables the IGRP routing protocol with the Autonomous System <i>100</i> .
<b>network</b> <i>10.0.0.0</i>	Advertises network 10.0.0.0 with IGRP.
<b>enable secret</b> <i>cisco</i>	Specifies an additional layer of security over the <b>enable password</b> command by displaying the encrypted <i>cisco</i> password.
<b>enable password</b> <i>san-fran</i>	Sets the local password to control access to various privilege levels to <i>san-fran</i> .
<b>interface e</b> <i>0/0</i>	Cisco 3640: most specify the slot, and then the interface.
<b>ip address</b> <i>address mask</i>	Let's pick 10.115.0.110 255.255.255.0.
<b>no shut</b>	Enables (activates) the interface.



## Task 2 Solution—Configuring the Serial Interface and Line

**Step 1** Configure the serial interface to which the external modem is connected and the corresponding line by using the following commands. In this case study, assume that a four-port synchronous/asynchronous network module is located in the fourth slot of a Cisco 3640, and that you are taking the first interface of that card for asynchronous communications.

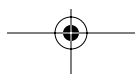
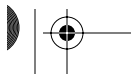
### NOTE

The modem is connected to the first port on this card, thus configuring **interface serial 3/0**. In ICRC and CRLS, you learn that Cisco starts counting at 0. Therefore the fourth slot is called slot 3 and the first port is called port 0—thus 3/0.

Command	Description
<b>interface serial 3/0</b>	Enters the interface configuration mode of the first interface of the fourth slot of the Cisco 3640.
<b>physical-layer async</b>	Configures the serial interface as an async interface (adds line 97 to the configuration).
<b>line 97</b>	Configures the line for the following physical layer parameters.
<b>login</b>	Allows login and challenge for a password.
<b>password cisco</b>	Sets the login password to <i>cisco</i> .
<b>modem inout</b>	Allows incoming and outgoing modem connections.
<b>transport input all</b>	Allows any transport protocol.
<b>speed 115200</b>	Sets speed between router and modem.
<b>stopbits 1</b>	One stop bit per byte.
<b>flowcontrol hardware</b>	Uses CTS/RTS flow control.

**Step 2** From global configuration mode, configure a hostname to allow reverse Telnet to the attached modem:

Command	Description
<b>ip host modem 2097 10.115.0.110</b>	Int E 0/0—IP address picked in Task 1. This command allows a reverse Telnet connection to line 97. The name can be anything you choose (we chose <i>modem</i> ). As learned previously in this chapter, a reverse Telnet connection is performed by using an IP address of a valid interface—in this case, interface E 0/0.



### Task 3 Solution—Configuring the Central Site Modem

- Step 1** Enter **modem** at the command prompt. This command (word: **modem**) connects you to the modem on line 97 of the Cisco 3640 via reverse Telnet, as specified in the previous step.
- Step 2** You are prompted for a Telnet line password. Enter the password **cisco**, and press Return.
- Step 3** Table 4-8 provides a description for each AT command. Results may vary, depending on the manufacturer.

**Table 4-8** *AT Commands to Manually Configure the Modem*

AT Command	Description
AT\$	HELP, Command Quick Reference (CTRL-S to Stop, CTRL-C to Cancel) <output omitted>
AT&\$	HELP, Ampersand Commands (CTRL-S to Stop, CTRL-C to Cancel) <output omitted>
AT\$\$	HELP, S Register Functions (CTRL-S to Stop, CTRL-C to Cancel) <output omitted>
AT&F1	Hardware Flow Control Configuration
ATS0=1	Auto-Answer on first ring
AT&C1	Modem Controls CD
AT&D2	DTE Controls DTR
AT&H1	CTS
AT&R2	RX to DTE/RTS high
AT&M4	ARQ/Normal Mode
AT&B1	Fixed DTE Speed
AT&K1	Auto Data Compression
AT&W0	Store Configuration to Template 0
ATI4	Displays the modem configuration settings

### Task 4 Solution—Configuring the PC Modem

From the PC, initiate a reverse Telnet session on the PC's modem. One way of doing this in Windows 95 is to use Hyperterminal and to establish a connection by selecting Connect using: Direct to COMX, where you will choose the COM port number used by your PC's modem.

After reverse Telnet is initiated, type **AT**; you should get an **OK** back from the modem. To call the Central site, type **atdt 1001**, which means "attention, dial tone the number 1001." Your PC modem will initiate a call on the Central site.

---

**NOTE** This case study was developed by using a POTS simulator, which used four-digit phone numbers.

---

Once the PC modem and the Central site modem are synchronized, log into the Central site by entering the line password you configured in Task 2. In the pcdemod window, you can enter the privileged EXEC mode. You can type **show users** to verify connectivity to the tty port 97. You can also type **show line 97** to verify asynchronous connectivity or type **show run** to display the configuration of the Central site.

```
CentralA#show run

!
version 11.3
no service password-encryption
!
hostname CentralA
!
enable secret 5 $1$kFDB$u7Gn10ROG1QUxISnDvEP1/
enable password san-fran
!
no ip domain-lookup (no misspelled command lookups)
ip host modem 2097 10.115.0.110 (static route to modem for reverse Telnet
connection)
!
controller T1 1/0
!
interface Ethernet0/0
ip address 10.115.0.110 255.255.255.0
no lat enabled
!
interface Ethernet0/1
no ip address
shutdown
no lat enabled
!
interface Serial3/0
physical-layer async (sets interface to async mode)
no ip address
!
interface Serial3/1
no ip address
shutdown
!
```

## Task 5 Solution—Simplifying Router Modem Configuration with Autoconfigure

**Step 1** From the privilege mode, enter **debug confmodem** to turn on modem configuration debugging.

**Step 2** Enter configuration mode and configure the modem line as follows:

```
line 97
modem autoconfigure type usr_sportster
```

In a few seconds, you should see messages from the command that you entered. Your output should look similar to the following:

```
TTY97: detection speed (115200) response ---OK---
TTY97: Modem command: --AT&F&C1&D2&H1&R2&M4&K1&B1S0=1H0--
TTY97: Modem configuration succeeded
TTY97: detection speed (115200) response ---OK---
TTY97: Done with modem configuration
```

**Step 3** To save your configuration, enter the following:

**CentralC#copy runn start**

## Case Study Conclusion

In this case study, you learned how to do the following:

- Configure a line on an access server
- Reverse Telnet to a modem
- Configure a modem
- Initiate a session on a modem
- Autoconfigure an access server's modem

## Case Study 4-2—Configuring Remote WAN Routers

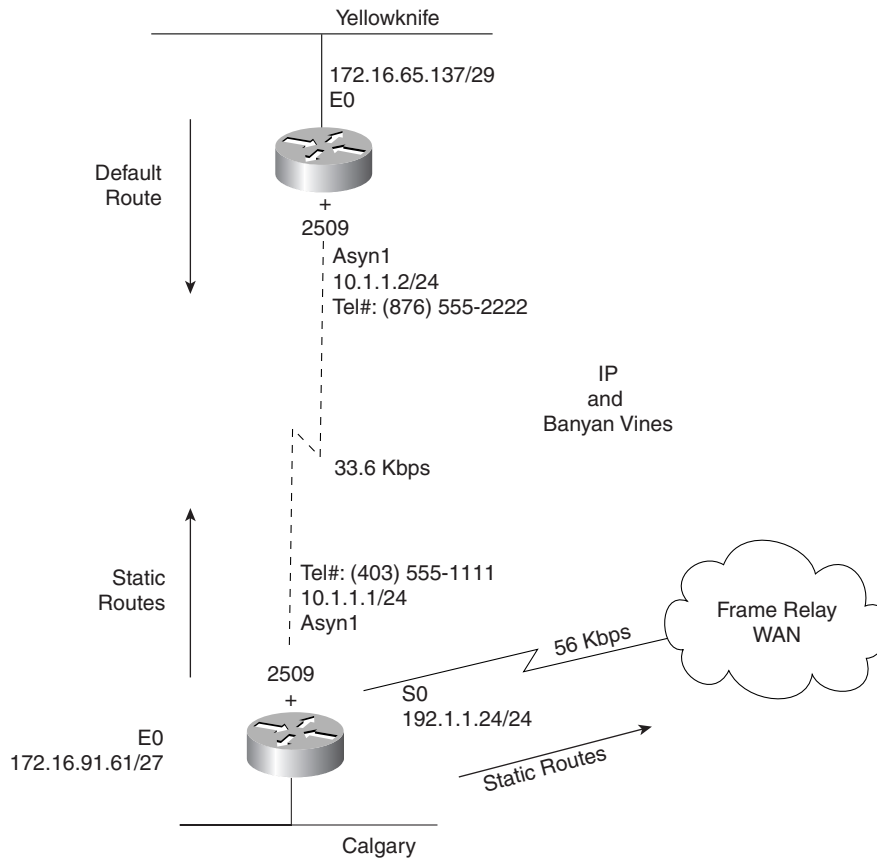
This case study presents a remote site that is located in the Northwest Territories that needs to connect to a cross-Canadian WAN.

### Scenario

Only analog lines are available for the Yellowknife customer. Therefore, Yellowknife will join the cross-Canadian WAN via Calgary's access server. Because the client has its own long-distance lines between Calgary and Yellowknife, it has decided to nail the line (permanent

connection via DDR). Figure 4-18 shows the topology of the WAN. The routers used in this case are Cisco 2509-RJ (2 serial, 1 Ethernet, and 8 Async).

**Figure 4-18** Remote Connection to a Cross-Canadian WAN



## Solution to Case Study 4-2—Configuring Remote WAN Routers

Here is the actual configuration of the Calgary and Yellowknife routers. The bold letters show the lines pertaining to the commands seen in this chapter:

```
Calgary#sh run
Building configuration...

Current configuration:
!
version 11.2
```

## 98 Chapter 4: Configuring Asynchronous Connections with Modems

---

```
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname Calgary
!
enable secret 5 $1$yZyJ$AT8R..8Agr2efa652aef6454sde
!
username Yellow password BABA
no ip domain-lookup
ip host Toronto 192.1.1.7
ip host Modem1 2001 172.16.171.161
ip host Halifax 192.1.1.46
ip host Montreal 192.1.1.3
ip host Ottawa 192.1.1.11
ip host Vancouver 192.1.1.26
ip host Edmonton 192.1.1.27
vines routing 4BADBEEF:0001
chat-script Hayes-Optima ABORT ERROR "" "AT Z" OK "ATDT \T" TIMEOUT 30 CONNECT \c
!
interface Ethernet0
  ip address 172.16.171.161 255.255.255.224
  vines metric 2
  vines propagate
  vines update interval 230
!
interface Serial0
  description DLCI-96 Frame Relay Circuit# 0B67851535-274TDON-352
  ip address 192.1.1.24 255.255.255.0
  encapsulation frame-relay
  vines metric 35
  vines propagate
  vines update interval 230
  no fair-queue
  cdp enable
!
interface Async1
  ip address 10.1.1.1 255.255.255.0
  encapsulation ppp
  keepalive 10
  async default routing
  async mode dedicated
  vines metric 50
  no vines propagate
  vines update interval 100
  dialer in-band
  dialer idle-timeout 300
  dialer wait-for-carrier-time 180
  dialer map vines 4BADCAFE:0001 name Yellow modem-script Hayes-Optima broadcast
  ➔9,18675552222
  dialer map ip 10.1.1.2 name Yellow modem-script Hayes-Optima broadcast
  ➔9,18675552222
  dialer-group 1
!
no ip classless
```

```
ip route 172.16.94.96 255.255.255.224 192.1.1.23
ip route 172.16.94.128 255.255.255.248 192.1.1.20
ip route 172.16.65.136 255.255.255.248 10.1.1.2
<output omitted>
!
snmp-server community public RO
dialer-list 1 protocol ip permit
dialer-list 1 protocol vines permit
banner incoming ^C
```

```
||
| Out of Bounds |
|
| Authorized Users Only |
^C
banner login ^C
```

```
||
| Out of Bounds |
|
| Authorized Users Only |
^C
banner motd ^Cncomm^C
!
line con 0
  exec-timeout 0 0
line 1
  login local
  modem InOut
  modem autoconfigure type hayes_optima
  transport input all
  stopbits 1
  speed 115200
  flowcontrol hardware
line 2 8
line aux 0
  password 7 YYYYYYYYYYYY
  login
  modem Dialin
  modem autoconfigure type usr_sportster
  stopbits 1
line vty 0 4
  password WWWWWWW
  login
!
end
```

```
Yellow#sh run
Building configuration...
```

```
Current configuration:
```

**100** Chapter 4: Configuring Asynchronous Connections with Modems

---

```
!  
version 11.2  
no service password-encryption  
no service udp-small-servers  
no service tcp-small-servers  
!  
hostname Yellow  
!  
enable secret 5 $&34%$Bea3..8Agr2efa65wf3f64ewe38$  
!  
username Calgary password BABA  
no ip domain-lookup  
ip host Modem1 2001 172.16.65.137  
ip host Calgary 10.1.1.1  
!  
interface Ethernet0  
 ip address 172.16.65.137 255.255.255.248  
 vines metric 30  
 no ip route-cache  
 no ip mroute-cache  
 no mop enabled  
!  
interface Serial0  
 no ip address  
 no ip route-cache  
 no ip mroute-cache  
 shutdown  
!  
interface Async1  
 ip address 10.1.1.2 255.255.255.0  
 encapsulation ppp  
 keepalive 10  
 async default routing  
 async mode dedicated  
 vines metric 50  
 no vines propagate  
 vines update interval 100  
 dialer in-band  
 dialer idle-timeout 300  
 dialer wait-for-carrier-time 180  
 dialer map vines 4BADBEEF:0001 name Calgary broadcast  
 dialer map ip 10.1.1.1 name Calgary broadcast  
 dialer-group 1  
!  
no ip classless  
ip route 0.0.0.0 0.0.0.0 10.1.1.1  
!  
dialer-list 1 protocol ip permit  
dialer-list 1 protocol vines permit  
!  
line con 0  
 exec-timeout 0 0  
line 1  
 login local  
 modem InOut
```

```
modem autoconfigure type hayes_optima
transport input all
stopbits 1
speed 115200
flowcontrol hardware
line 2 8
line aux 0
session-timeout 15
password cisco
autobaud
login
modem InOut
transport input all
stopbits 1
line vty 0 4
password cisco
login
!
end
```

## Solution Summary

On this router, I configured the interface Async; not Serial. The access server used by Calgary is a Cisco 2509-RJ, which has built-in async interfaces.

Even though the modem attached to the access server was a Hayes Accura 33.6, I had to use the Hayes Optima modemcap to get the connection to work. Also, a chat-script had to be configured for dial-out to take place.

## Review Questions

Answer the following questions and then refer to Appendix H, "Answers to Review Questions," for answers and explanations.

- 1 What does the lock DTE modem attribute do?
- 2 A user dials into a line and ends up in someone else's session. What is one possible cause?
- 3 What is reverse Telnet? Describe how it is used with modems.
- 4 What are the modem autoconfiguration options?