Chapter 11

Communications and Networking

Most computer-to-computer connections occur through a serial port, a parallel port, or a network adapter. In this chapter, you explore ways to connect your PC to other computers. Such connections enable you to transfer and share files, send electronic mail, access software on other computers, and generally make two or more computers behave as a team.

Using Communications Ports and Devices

The basic communications ports in any PC system are the serial and parallel ports. The serial ports are used primarily for devices that must communicate bidirectionally with the system; such devices include modems, mice, scanners, digitizers, or any other device that "talks to" and receives information from the PC.

Parallel ports are used primarily for printers and operate normally as one-way ports, although sometimes they can be used bidirectionally. Several companies also manufacture communications programs that perform high-speed transfers between PC systems using serial or parallel ports. Several products currently on the market make untraditional use of the parallel port. You can purchase network adapters, floppy disk drives, or tape backup units that use the parallel port, for example.

Serial Ports

The asynchronous serial interface is the primary system-to-system communications device. *Asynchronous* means that no synchronization or clocking signal is present, so characters may be sent with any arbitrary time spacing, as when a typist is providing the data.

Each character is framed by a standard start and stop signal. A single 0 bit, called the *start bit*, precedes each character to tell the receiving system that the next eight bits constitute a byte of data. One or two stop bits follow the character to signal that the character has been sent. At the receiving end of the communication, characters are recognized by the start and stop signals instead of being recognized by the timing of their arrival. The asynchronous interface is character-oriented and has about 20 percent overhead for the extra information needed to identify each character.

Serial refers to data sent over one wire, with each bit lining up in a series as the bits are sent. This type of communication is used over the phone system, because this system provides one wire for data in each direction. Add-on serial ports for the PC are available from many manufacturers. You usually can find these ports on one of the available multifunction boards or on a board with at least a parallel port. Figure 11.1 shows the

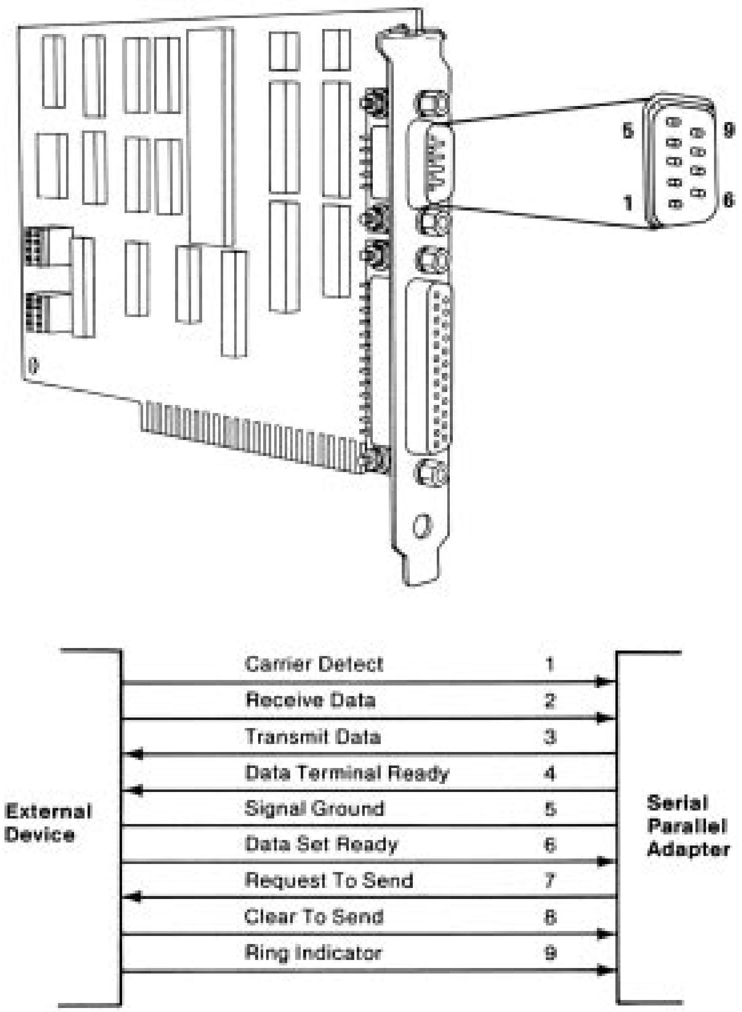
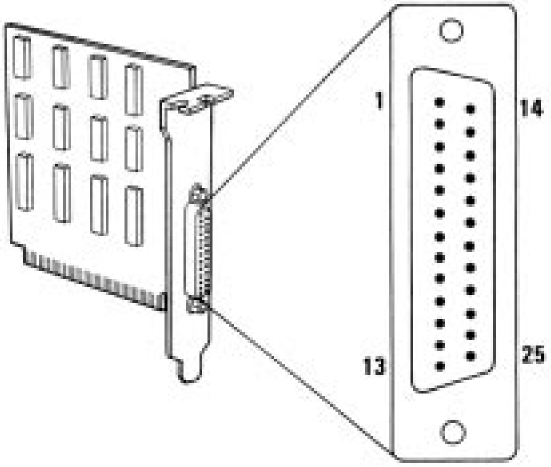
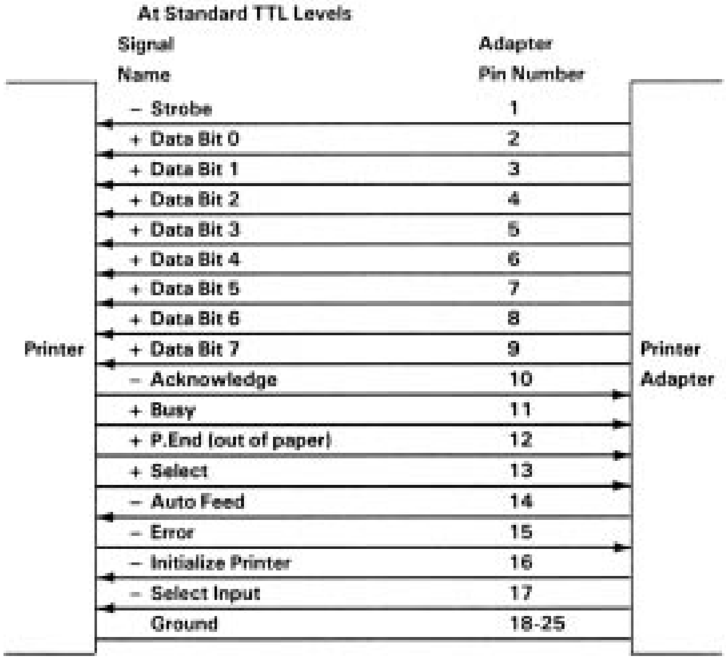


Fig. 11.1

AT-style 9-pin serial-port connector specifications.





Signal Name	Adapter Pin Number
− Strobe	1
+ Data Bit 0	2
+ Data Bit 1	3
+ Data Bit 2	4
+ Data Bit 3	5
+ Data Bit 4	6
+ Data Bit 5	7
+ Data Bit 6	8
+ Data Bit 7	9
− Acknowledge	10
+ Busy	11
+ P.End (out of paper)	12
+ Select	13
− Auto Feed	14
− Error	15
− Initialize Printer	16
− Select Input	17
Ground	18-25

standard 9-pin
AT-style serial port, and figure 11.2 shows the more conventional 25-pin version.

Fig. 11.2
Standard 25-pin serial-port connector specifications.

Serial ports may connect to a variety of devices such as modems, plotters, printers, other computers, bar code readers, scales, and device control circuits. Basically, anything that needs a two-way connection to the PC uses the industry-standard Reference Standard number 232 revision c (RS-232c) serial port. This device enables data transfer between otherwise incompatible devices. Tables 11.1, 11.2, and 11.3 show the pinouts of the 9-pin (AT-style), 25-pin, and 9-pin-to-25-pin serial connectors.

Table 11.1 9-Pin (AT) Serial Port Connector

Pin	Signal	Description	I/O
1	CD	Carrier detect	In
2	RD	Receive data	In
3	TD	Transmit data	Out
4	DTR	Data terminal ready	Out
5	SG	Signal ground	—
6	DSR	Data set ready	In
7	RTS	Request to send	Out
8	CTS	Clear to send	In
9	RI	Ring indicator	In

Table 11.2 25-Pin (PC, XT, and PS/2) Serial Port Connector

Pin	Signal	Description	I/O
1	—	Chassis ground	—
2	TD	Transmit data	Out
3	RD	Receive data	In
4	RTS	Request to send	Out
5	CTS	Clear to send	In
6	DSR	Data set ready	In
7	SG	Signal ground	—
8	CD	Carrier detect	In
9	—	+Transmit current loop return	Out
11	—	–Transmit current loop data	Out
18	—	+Receive current loop data	In
20	DTR	Data terminal ready	Out
22	RI	Ring indicator	In
25	—	–Receive current	

		loop return	In

Pins 9, 11, 18, and 25 are used for a Current Loop interface only. Current Loop is not supported on the AT Serial/Parallel Adapter or PS/2 systems.

Table 11.3 9-Pin to 25-Pin Serial Cable Adapter Connections

9-Pin	25-Pin	Signal	Description
1	8	CD	Carrier detect
2	3	RD	Receive data
3	2	TD	Transmit data
4	20	DTR	Data terminal ready
5	7	SG	Signal ground
6	6	DSR	Data set ready
7	4	RTS	Request to send
8	5	CTS	Clear to send
9	22	RI	Ring indicator

UART Chips. The heart of any serial port is the Universal Asynchronous Receiver/Transmitter (UART) chip. This chip competely controls breaking the native parallel data within the PC into serial format, and later converting serial data back into the parallel format.

There are several types of UART chips on the market. The original PC and XT used the 8250 UART, which still is used in many low-price serial cards on the market. In the PC/AT (or other systems based on at least an 80286), the 16450 UART is used. The only difference between these chips is their suitability for high-speed communications. The 16450 is better suited for high-speed communications than the 8250; otherwise, both chips appear identical to most software.

The 16550 UART was the first serial chip used in the PS/2 line. This chip could function as the earlier 16450 and 8250 chips, but it also included a 16-byte buffer that aided in faster communications. Unfortunately, the 16550 also had a few bugs, particularly in the buffer area. These bugs were corrected with the release of the 16550A UART, which is used in all high-performance serial ports.

Because the 16550A is a faster, more reliable chip than its predecessors, it is best to look for serial ports that use it. If you are in doubt about which chip you have in your system, you can use the Microsoft MSD program (provided with Windows or DOS 6) to determine the type of UART you have.

Most UART chips used by IBM are made by National Semiconductor. You also can identify the chips by looking for the largest chip on the serial port card and reading the numbers on that chip. Usually the chips are socketed, and replacing only the chip may be possible. Table 11.4 provides a complete list of UART chips that may be in your system.

Note

The interrupt bug referred to in table 11.4 is a spurious interrupt generated by the 8250 at the end of an access. The ROM BIOS code in the PC and XT has been written to work around this bug. If a chip without the bug is installed, random lockups may occur. The 16450 or 16550(A) chips do not have the interrupt bug, and the AT ROM BIOS was written without any of the bug workarounds in PC or XT systems.

Table 11.4 UART Chips in PC or AT Systems

Chip	Description
8250	IBM used this original chip in the PC serial port card. The chip has several bugs, none of which is serious. The PC and XT ROM BIOS are written to anticipate at least one of the bugs. This chip was replaced by the 8250B.
8250A	Do not use the second version of the 8250 in any system. This upgraded chip fixes several bugs in the 8250, including one in the interrupt enable register, but because the PC and XT ROM BIOS expect the bug, this chip does not work properly with those systems. The 8250A should work in an AT system that does not expect the bug, but does not work adequately at 9600 bps.
8250B	The last version of the 8250 fixes bugs from the previous two versions. The interrupt enable bug in the original 8250, expected by the PC and XT ROM BIOS software, has been put back into this chip, making the 8250B the most desirable chip for any non-AT serial port application. The 8250B chip may work in an AT under DOS, but does not run properly at 9600 bps.
16450	IBM selected the higher-speed version of the 8250 for the AT. Because this chip has fixed the interrupt enable bug mentioned earlier, the 16450 does not operate properly in many PC or XT systems because they expect this bug to be present. OS/2 requires this chip as a minimum, or the serial ports do not function properly. It also adds a scratch-pad register as the highest register. The 16450 is used primarily in AT systems because of its increase in throughput over the 8250B.
16550	This newer UART improves on the 16450. This chip cannot be used in a FIFO (first in, first out) buffering mode because of problems with the design, but it does enable a programmer to use multiple DMA channels and thus increase throughput on an AT or higher class computer system. I highly recommend replacing the 16550 UART with the 16550A.
16550A	This chip is a faster 16450 with a built-in 16-character Transmit and Receive FIFO (first in, first out) buffer that works. It also allows multiple DMA channel access. You should install this chip in your AT system serial port cards if you do any serious communications at 9600 bps or higher. If your communications program makes use of the FIFO, which most do today, it can greatly increase communications speed and eliminate lost characters and data at the higher speeds.

Various manufacturers make versions of the 16550A; National Semiconductor was the first. Its full part number for the 40-pin DIP is NS16550AN or NS16550AFN. Make sure that the part you get is the 16550A, and not the older 16550. You can contact Fry's Electronics or Jameco Electronics to obtain the NS16550AN, for example.

Serial-Port Configuration

Each time a character is received by a serial port, it has to get the attention of the computer by raising an Interrupt Request Line (IRQ). Eight-bit ISA bus systems have eight of these lines, and systems with a 16-bit ISA bus have 16 lines. The 8259 interrupt controller chip usually handles these requests for attention. In a standard configuration, COM1 uses IRQ4, and COM2 uses IRQ3.

When a serial port is installed in a system, it must be configured to use specific I/O addresses (called ports), and interrupts (called IRQs for Interrupt ReQuest). The best plan is to follow the existing standards for how these devices should be set up. For configuring serial ports, you should use the addresses and interrupts indicated in table 11.5.

Table 11.5 Standard Serial I/O Port Addresses and Interrupts

System	COMx	Port	IRQ
All	COM1	3F8h	IRQ4
All	COM2	2F8h	IRQ3
ISA bus	COM3	3E8h	IRQ4
ISA bus	COM4	2E8h	IRQ3
ISA bus	COM3	3E0h	IRQ4
ISA bus	COM4	2E0h	IRQ3
ISA bus	COM3	338h	IRQ4
ISA bus	COM4	238h	IRQ3
MCA bus	COM3	3220h	IRQ3
MCA bus	COM4	3228h	IRQ3
MCA bus	COM5	4220h	IRQ3
MCA bus	COM6	4228h	IRQ3
MCA bus	COM7	5220h	IRQ3
MCA bus	COM8	5228h	IRQ3

A problem can occur when the ROM BIOS logs in these ports. If the Power-On Self-Test (POST) does not find a 3F8 serial port but does find a 2F8, then the 2F8 serial port is mistakenly assigned to COM1. The reserved IRQ line for COM1 is IRQ4, but this serial port of 2F8 is using COM2's address, which means that it should be using IRQ3 instead of IRQ4. If you are trying to use BASIC or DOS for COM1 operations, therefore, the serial port or modem cannot work.

Another problem is that IBM never built BIOS support in its original ISA bus systems for COM3 and COM4. Therefore, the DOS MODE command cannot work with serial ports above COM2 because DOS gets its I/O information from the BIOS, which finds out what is installed and where in your system during the POST. The POST in these older systems checks only for the first two installed ports. PS/2 systems have an improved BIOS that checks for as many as eight serial ports, although DOS is limited to handling only four of them.

To get around this problem, most communications software and some serial peripherals (such as mice) support higher COM ports by addressing them directly, rather than making DOS function calls. The communications program PROCOMM, for example, supports the additional ports even if your BIOS or DOS does not. Of course, if your system or software does not support these extra ports or you need to redirect data using the MODE command, trouble arises.

A couple of utilities enable you to append your COM port information to the BIOS, making the ports DOS-accessible. A program called Port Finder is one of the best, and is available in the “general hardware” data library of the IBMHW forum on CompuServe.

Port Finder activates the extra ports by giving the BIOS the addresses and providing utilities for swapping the addresses among the different ports. Address-swapping enables programs that don’t support COM3 and COM4 to access them. Software that already directly addresses these additional ports usually is unaffected.

Extra ports, however, must use separate interrupts. If you are going to use two COM ports at one time, they should be on opposite interrupts. Using the standard port and interrupt configurations, the possibilities for simultaneous operation follow:

COM1 (IRQ4) and COM2 (IRQ3)

COM1 (IRQ4) and COM4 (IRQ3)

COM2 (IRQ3) and COM3 (IRQ4)

COM3 (IRQ4) and COM4 (IRQ3)

Divide your COM port inputs into these groups of two, pairing serial devices that will not be used simultaneously on the same interrupt, and devices that will be used at the same time on different interrupts. Note again that PS/2 Micro Channel Architecture systems are entirely exempt from these types of problems because they have a BIOS that looks for the additional ports, and because the MCA bus can share interrupts without conflicts.

To configure serial boards in ISA bus systems, you probably will have to set jumpers and switches. Because each board on the market is different, you always should consult the OEM manual for that particular card if you need to know how the card should or can be configured. IBM includes this information with each card’s documentation. IBM also offers technical-reference options and adapters manuals as well as hardware-maintenance and service manuals, which also cover in detail all the jumper and switch settings for IBM adapter cards. Other manufacturers simply include with the card a manual that describes the card and includes the configuration information. The PS/2

MCA bus systems have an automatic or software-driven configuration that eliminates these configuration hassles.

Modem Standards

Bell Labs and the CCITT have set standards for modem protocols. Although the CCITT is actually a French term, it translates into English as the Consultative Committee on International Telephone and Telegraph. A *protocol* is a method by which two different entities agree to communicate. Bell Labs no longer sets new standards for modems, although several of its older standards still are used. Most modems built in the last few years conform to the CCITT standards. The CCITT is an international body of technical experts responsible for developing data communications standards for the world. The group falls under the organizational umbrella of the United Nations, and its members include representatives from major modem manufacturers, common carriers (such as AT&T), and governmental bodies.

The CCITT establishes communications standards and protocols in many areas, so one modem often adheres to several CCITT standards, depending on its various features and capabilities. Modem standards can be grouped into the following three areas:

- Modulation standards

  Bell 103
  Bell 212A
  CCITT V.21
  CCITT V.22bis
  CCITT V.29
  CCITT V.32
  CCITT V.32bis

- Error-correction standards

  CCITT V.42

- Data-compression standards

  V.42bis

Other standards have been developed by different companies, and not Bell Labs or the CCITT. These are sometimes called *proprietary standards*, even though most of these companies publish full specifications on their protocols so that other manufacturers can develop modems to work with them. The following list shows some of the proprietary standards that have become fairly popular:

- Modulation

  HST
  PEP
  DIS

- Error correction

  - MNP 1-4
  - Hayes V-series
- Data compression
  - MNP 5
  - CSP

Almost all modems today claim to be *Hayes compatible*, which does not refer to any communication protocol, but instead to the commands required to operate the modem. Because almost every modem uses the Hayes command set, this compatibility is a given and should not really affect your decisions about modems. Table 11.6 lists the command sets for the U.S. Robotics and Hayes brands of modems.

Table 11.6 U.S. Robotics and Hayes Modem Commands and Supported Features

Command	Modem Functions and Options	USR Dual	2400	Hayes 2400	1200
&	See Extended Command Set	×			
%	See Extended Command Set	×			
A	Force Answer mode when modem has not received an incoming call	×	×	×	×
A/	Re-execute last command once	×	×	×	×
A>	Repeat last command continuously	×			
Any key	Terminate current connection attempt; exit Repeat mode	×	×		
AT	Attention: must precede all other commands except A/, A>, and +++	×	×	×	×
BN	Handshake options	×		×	
	BO CCITT answer sequence	×		×	
	B1 Bell answer tone	×		×	
Cn	Transmitter On/Off	×	×	×	×
	C0 Transmitter Off	×	×	×	×
	C1 Transmitter On-Default	×	×	×	×
Dn	Dial number n and go into originate mode. Use any of these options:	×	×	×	×
	P Pulse dial-Default	×	×	×	×
	T Touch-Tone dial	×	×	×	×
	, (Comma) Pause for 2 seconds	×	×	×	×
	; Return to command state after dialing	×	×	×	×
	"… Dial the letters that follow	×	×		
	! Flash switch-hook to transfer call	×	×	×	

	W	Wait for 2nd dial tone (if X3 or higher is set)	×	×	×	
	@	Wait for an answer (if X3 or higher is set)	×	×	×	
	R	Reverse frequencies	×	×	×	×
	S	Dial stored number			×	
DL	Dial the last-dialed number		×			

Command	**Modem Functions and Options**		**USR Dual**	**2400**	**Hayes 2400**	**1200**
DSn	Dial number stored in NVRAM at position n		×			
En	Command mode local echo; not applicable after a connection has been made		×	×	×	×
	E0	Echo Off	×	×	×	×
	E1	Echo On	×	×	×	×
Fn	Local echo On/Off when a connection has been made		×	×	×	×
	F0	Echo On (Half duplex)	×	×	×	×
	F1	Echo Off (Full duplex)-Default	×	×	×	×
Hn	On/Off hook control		×	×	×	×
	H0	Hang up (go on hook)-Default	×	×	×	×
	H1	Go off hook	×	×	×	×
In	Inquiry		×	×	×	×
	I0	Return product code	×	×	×	×
	I1	Return memory (ROM) checksum	×	×	×	×
	I2	Run memory (RAM) test	×	×	×	
	I3	Return call duration/real time	×	×		
	I4	Return current modem settings	×	×		
	I5	Return NVRAM settings	×			
	I6	Return link diagnostics	×			
	I7	Return product configuration	×			
Kn	Modem clock operation		×			
	KO	ATI3 displays call duration-Default	×			
	K1	ATI3 displays real time; set with ATI3=HH:MM:SSK1	×			
Ln	Loudness of speaker volume;				×	
	L0	Low			×	
	L1	Low			×	
	L2	Medium			×	
	L3	High			×	
Mn	Monitor (speaker) control		×	×	×	×
	M0	Speaker always Off	×	×	×	×
	M1	Speaker On until carrier is established-	×	×	×	×

(continues)

	Default				
	M2 Speaker always On	×	×	×	×
	M3 Speaker On after last digit dialed, Off at carrier detect	×	×	×	×
O	Return on-line after command execution	×	×	×	×
	O0 Return on-line, normal	×	×	×	×
	O1 Return on-line, retrain	×	×	×	×

Table 11.6 Continued

		USR		Hayes	
Command	**Modem Functions and Options**	**Dual**	**2400**	**2400**	**1200**
P	Pulse dial	×	×	×	×
Qn	Result codes display	×	×	×	×
	Q0 Result codes displayed	×	×	×	×
	Q1 Result codes suppressed (quiet mode)	×	×	×	×
	Q2 Quiet in answer mode only	×			
Sr=n	Set register commands: r is any S-register; n must be a decimal number between 0 and 255.	×	×	×	×
Sr.b=n	Set bit .b of register r to n (0/Off or 1/On)	×			
Sr?	Query register r	×	×	×	×
T	Tone dial	×	×	×	×
Vn	Verbal/Numeric result codes	×	×	×	×
	V0 Numeric mode	×	×	×	×
	V1 Verbal mode	×	×	×	×
Xn	Result code options	×	×	×	×
Yn	Long space disconnect			×	
	Y0 Numeric mode			×	
	Y1 Enabled; disconnects after 1.5-second break			×	
Z	Software reset	×	×	×	×
+++	Escape code sequence, preceded and followed by at least one second of no data transmission	×	×		
/(Slash)	Pause for 125 msec	×			
>	Repeat command continuously or up to 10 dial attempts; cancel by pressing any key	×	×		
$	Online Help - Basic command summary	×	×		
&$	Online Help - Ampersand command summary	×			
%$	Online Help - Percent command summary	×			
D$	Online Help - Dial command summary	×	×		
S$	Online Help - S-register summary	×	×		
<Ctrl>-S	Stop/restart display of HELP screens		×		
<Ctrl>-C	Cancel display HELP screens		×		

<Ctrl>-K	Cancel display HELP screens		×		
Extended Command Set					
&An	ARQ result codes 14-17, 19	×			
	&A0 Supppress ARQ result codes	×			
	&A1 Display ARQ result codes-Default	×			
	&A2 Display HST and V.32 result codes	×			
	&A3 Display protocol result codes	×			

Command	Modem Functions and Options	USR Dual	2400	Hayes 2400	1200
Extended Command Set					
&Bn	Data Rate, terminal-to-modem (DTE/DCE)	×			
	&B0 DTE rate follows connection rate-Default	×			
	&B1 Fixed DTE rate	×			
	&B2 Fixed DTE rate in ARQ mode; variable DTE rate in non-ARQ mode	×			
&Cn	Carrier Detect (CD) operations	×		×	
	&C0 CD override	×		×	
	&C1 Normal CD operations	×		×	
&Dn	Data Terminal Ready (DTR) operations	×		×	
	&D0 DTR override	×		×	
	&D1 DTR Off; goes to command state			×	
	&D2 DTR Off; goes to command state and on hook	×		×	
	&D3 DTR Off; resets modem			×	
&F	Load factory settings into RAM	×		×	
&Gn	Guard tone	×		×	
	&G0 No guard tone; U.S., Canada-Default	×		×	
	&G1 Guard tone; some European countries	×		×	
	&G2 Guard tone; U.K., requires B0	×		×	
&Hn	Transmit Data flow control	×			
	&H0 Flow control disabled-Default	×			
	&H1 Hardware (CTS) flow control	×			
	&H2 Software (XON/XOFF) flow control	×			
	&H3 Hardware and software control	×			
&In	Received Data software flow control	×			
	&I0 Flow control disabled-Default	×			
	&I1 XON/XOFF to local modem and remote computer	×			
	&I2 XON/XOFF to local modem only	×			

(continues)

	&I3 Host mode, Hewlett-Packard protocol	×			
	&I4 Terminal mode, Hewlett-Packard protocol	×			
	&I5 ARQ mode-same as &I2; non-ARQ mode; look for incoming XON/XOFF	×			
&Jn	Telephone Jack selection			×	
	&J0 RJ-11/RJ-41S/RJ-45S			×	
	&J1 RJ-12/RJ-13			×	

Table 11.6 Continued

Command	Modem Functions and Options	USR Dual	2400	Hayes 2400	1200
Extended Command Set					
&Kn	Data compression	×			
	&K0 Disabled	×			
	&K1 Auto enable/disable-Default	×			
	&K2 Enabled	×			
	&K3 V.42bis only	×			
&Ln	Normal/Leased line operation	×		×	
	&L0 Normal phone line-Default	×		×	
	&L1 Leased line	×		×	
&Mn	Error Control/Synchronous Options	×		×	
	&M0 Normal mode, no error control	×		×	
	&M1 Synch mode	×		×	
	&M2 Synch mode 2 - stored number dialing			×	
	&M3 Synch mode 3 - manual dialing	×			
	&M4 Normal/ARQ mode-Normal if ARQ connection cannot be made-Default	×			
	&M5 ARQ mode-hang up if ARQ connection cannot be made	×			
&Nn	Data Rate, data link (DCE/DCE)	×			
	&N0 Normal link operations-Default	×			
	&N1 300 bps	×			
	&N2 1200 bps	×			
	&N3 2400 bps	×			
	&N4 4800 bps	×			
	&N5 7200 bps	×			
	&N6 9600 bps	×			
	&N7 12K bps	×			
	&N8 14.4K bps	×			

&Pn	Pulse dial make/break ratio	×		×	
	&P0 North America-Default	×		×	
	&P1 British Commonwealth	×		×	
&Rn	Received Data hardware (RTS) flow control	×		×	
	&R0 CTS tracks RTS	×		×	
	&R1 Ignore RTS-Default	×		×	
	&R2 Pass received data on RTS high; used Pass received data on RTS high; used	×			

Command	Modem Functions and Options	USR Dual	2400	Hayes 2400	1200
Extended Command Set					
&Sn	Data Set Ready (DSR) override	×		×	
	&S0 DSR override (always On-Default)	×		×	
	&S1 Modem controls DSR	×		×	
	&S2 Pulsed DSR; CTS follows CD	×			
	&S3 Pulsed DSR	×			
&Tn	Modem Testing	×		×	
	&T0 End testing	×		×	
	&T1 Analog loopback	×		×	
	&T2 Reserved	×			
	&T3 Digital loopback	×		×	
	&T4 Grant remote digital loopback	×		×	
	&T5 Deny remote digital loopback	×		×	
	&T6 Initiate remote digital loopback	×		×	
	&T7 Remote digital loopback with self test	×		×	
	&T8 Analog loopback with self test	×		×	
&W	Write current settings to NVRAM	×		×	
&×n	Synchronous timing source	×		×	
	&X0 Modem's transmit clock-Default	×		×	
	&X1 Terminal equipment	×		×	
	&X2 Modem's receiver clock	×		×	
&Yn	Break handling. Destructive breaks clear the buffer; expedited breaks are sent immediately to remote system.	×			
	&Y0 Destructive, but don't send break	×			
	&Y1 Destructive, expedited-Default	×			
	&Y2 Nondestructive, expedited	×			
	&Y3 Nondestructive, unexpedited	×			
&Zn=L	Store last-dialed phone number in NVRAM at position n	×			

(continues)

&Zn=s	Write phone number(s) to NVRAM at position n (0-3); 36 characters maximum	×			
&Zn?	Display phone number in NVRAM at position n (n=0-3)	×		×	
%Rn	Remote access to Rack Controller Unit (RCU)	×			
	%R0 Disabled	×			
	%R1 Enabled	×			
%T	Enable Touch-Tone recognition	×			

Table 11.6 Continued

Command	Modem Functions and Options	USR Dual	2400	Hayes 2400	1200
Modem S-Register Functions and Defaults					
SO	Number of rings before automatic answering when DIP switch 5 is UP. Default = 1. SO = 0 disables Auto Answer, equivalent to DIP switch 5 Down	SW5	SW5	0	SW5
S1	Counts and stores number of rings from incoming call	0	0	0	0
S2	Define escape code character. Default = +.	43	43	43	43
S3	Define ASCII carriage return	13	13	13	13
S4	Define ASCII line feed	10	10	10	10
S5	Define ASCII Backspace	8	8	8	8
S6	Number of seconds modem waits before dialing	2	2	2	2
S7	Number of seconds modem waits for a carrier	60	30	30	30
S8	Duration (sec) for pause (,) option in Dial command and pause between command reexecutions for Repeat (>) command	2	2	2	2
S9	Duration (.1 sec units) of remote carrier signal before recognition	6	6	6	6
S10	Duration (.1 sec units) modem waits after loss of carrier before hanging up	7	7	7	7
S11	Duration and spacing (ms) of dialed Touch-Tones	70	70	70	70
S12	Guard time (in .02 sec units) for escape code sequence	50	50	50	50
S13	Bit-mapped register:	0			
	1 Reset when DTR drops				
	2 Auto answer in originate mode				
	4 Disable result code pause				
	8 DS0 on DTR low-to-high				
	16 DS0 on power up, ATZ				

Command	Modem Functions and Options		USR Dual	2400	Hayes 2400	1200
	32	Disable HST modulation				
	64	Disable MNP Level 3				
	128	Watchdog hardware reset				
S15	Bit-mapped register:		0			
	1	Disable high-frequency equalization				
	2	Disable on-line fallback				
	4	Force 300-bps back channel				
	8	Set non-ARQ transmit buffer to 128 bytes				
Modem S-Register Functions and Defaults						
	16	Disable MNP Level 4				
	32	Set Del as Backspace key				
	64	Unusual MNP incompatibilty				
	128	Custom applications only				
S16	Bit-mapped register:		0	0	0	
	1	Analog loopback				
	2	Dial test				
	4	Test pattern				
	8	Initiate remote digital loopback				
	16	Reserved				
	32	Reserved				
	64	Reserved				
	128	Reserved				
S18	&Tn Test timer, disabled when set to 0		0		0	
S19	Set inactivity timer in minutes		0			
S21	Length of Break, DCE to DTE, in 10ms units		10		0	
S22	Define ASCII XON		17		17	
S23	Define ASCII XOFF		19		19	
S24	Duration (20ms units) of pulsed DSR when modem is set to &S2 or &S3		150			
S25	Delay to DTR		5			
S26	Duration (10ms units) of delay between RTS and CTS, synchronous mode		1		1	
S27	Bit-mapped register:		0			
	1	Enable V.21 modulation, 300 bps				

(continues)

	2 Enable unencoded V.32 modulation				
	4 Disable V.32 modulation				
	8 Disable 2100 Hz answer tone				
	16 Disable MNP handshake				
	32 Disable V.42 Detect phase				
	64 Reserved				
	128 Unusual software incompatibility				
S28	Duration (.1 sec units) of V.21/V.23 handshake delay	8			

Table 11.6 Continued

Command	Modem Functions and Options	USR Dual	2400	Hayes 2400	1200
Modem S-Register Functions and Defaults					
S32	Voice/Data switch options:	1			
	0 Disabled				
	1 Go off hook in originate mode				
	2 Go off hook in answer mode				
	3 Redial last-dialed number				
	4 Dial number stored at position 0				
	5 Auto answer toggle On/Off				
	6 Reset modem				
	7 Initiate remote digital loopback				
S34	Bit-mapped register:	0			
	1 Disable V.32bis				
	2 Disable enhanced V.32 mode				
	4 Disable quick V.32 retrain				
	8 Enable V.23 modulation				
	16 Change MR LED to DSR				
	32 Enable MI/MIC				
	64 Reserved				
	128 Reserved				
S38	Duration (sec) before disconnect when DTR drops during an ARQ call	0			

ARQ = Automatic repeat request
ASCII = American Standard Code for Information Interchange
BPS = Bits per second
CCITT = Consultative Committee for International Telephone and Telegraph
CD = Carrier detect
CRC = Cyclic redundancy check
DCE = Data communications equipment
DTE = Data terminal equipment
EIA = Electronic Industries Association
HDLC = High-level data link control
HST = High-speed technology

Hz = Hertz
LAPM = Link access procedure for modems
MI/MIC = Mode indicate/Mode indicate common
MNP = Microcom networking protocol
NVRAM = Non-volatile memory
RAM = Random-access memory
ROM = Read-only memory
SDLC = Synchronous Data Link Control
MR = Modem ready
LED = Light-emitting diode
DTR = Data terminal ready
CTS = Clear to send
RTS = Ready to send
DSR = Data Set Ready

Modulation Standards. Modems start with *modulation*, which is the electronic signaling method used by the modem (from modulator to demodulator). Modems must use the same modulation method to understand each other. Each data rate uses a different modulation method, and sometimes more than one method exists for a particular rate.

The three most popular modulation methods are frequency-shift keying (FSK), phase-shift keying (PSK), and quadrature-amplitude modulation (QAM). FSK is a form of frequency modulation, otherwise known as FM. By causing and monitoring frequency changes in a signal sent over the phone line, two modems can send information. PSK is a form of phase modulation, in which the timing of the carrier signal wave is altered and the frequency stays the same. QAM is a modulation technique that combines phase changes with signal-amplitude variations, resulting in a signal that can carry more information than the other methods.

Baud versus Bits Per Second (bps). Baud rate and the bit rate often are confused in discussions about modems. *Baud rate* is the rate at which a signal between two devices changes in one second. If a signal between two modems can change frequency or phase at a rate of 300 times per second, for example, that device is said to communicate at 300 baud. Sometimes a single modulation change is used to carry a single bit. In that case, 300 baud also equals 300 bits per second (bps). If the modem could signal two bit values for each signal change, the bit-per-second rate would be twice the baud rate, or 600 bps at 300 baud. Most modems transmit several bits per baud, so that the actual baud rate is much slower than the bit-per-second rate. In fact, people usually use the term *baud* incorrectly. We normally are not interested in the raw baud rate, but in the bit-per-second rate, which is the true gauge of communications speed.

Bell 103. Bell 103 is a U.S. and Canadian 300-bps modulation standard. It uses frequency-shift-keying (FSK) modulation at 300 baud to transmit one bit per baud. Most higher-speed modems will communicate using this protocol, even though it is largely obsolete.

Bell 212A. Bell 212A is the U.S. and Canadian 1200-bps modulation standard. It uses differential phase-shift keying (DPSK) at 600 baud to transmit two bits per baud.

V.21. V.21 is an international data-transmission standard for 300-bps communications similar to Bell 103. Because of some differences in the frequencies used, Bell 103 modems are not compatible with V.21 modems. This standard is used primarily outside the United States.

V.22. V.22 is an international 1200-bps data-transmission standard. This standard is similar to the Bell 212A standard, but is incompatible in some areas, especially in answering a call. This standard was used primarily outside the United States.

V.22bis. V.22bis is a data-transmission standard for 2400-bps communications. *Bis* is Latin for *second*, indicating that this data transmission is an improvement to or follows V.22. This data transmission is an international standard for 2400 bps and is used inside and outside the United States. V.22bis uses quadrature-amplitude modulation (QAM) at 600 baud and transmits four bits per baud to achieve 2400 bps.

V.23. V.23 is a split data-transmission standard, operating at 1200 bps in one direction and 75 bps in the reverse direction. Therefore, the modem is only *pseudo-full-duplex*, meaning that it can transmit data in both directions simultaneously, but not at the maximum data rate. This standard was developed to lower the cost of 1200-bps modem technology, which was expensive in the early 1980s. This standard was used primarily in Europe.

V.29. V.29 is a data-transmission standard at 9600 bps, which defines a half duplex (one-way) modulation technique. This standard generally is used in Group III facsimile (fax) transmissions, and only rarely in modems. Because V.29 is a half-duplex method, it is substantially easier to implement this high-speed standard than to implement a high-speed full-duplex standard. As a modem standard, V.29 has not been fully defined, so V.29 modems of different brands seldom can communicate with each other. This does not affect fax machines, which have a fully defined standard.

V.32. V.32 is a full-duplex (two-way) data transmission standard at 9600 bps. It is a full modem standard, and also includes forward error-correcting and negotiation standards. V.32 uses TCQAM (trellis coded quadrature amplitude modulation) at 2400 baud to transmit 4 bits per baud, resulting in the 9600-bps transmission speed. The trellis coding is a special forward error-correction technique that creates an additional bit for each packet of 4. This extra check bit is used to enable on-the-fly error correction to take place at the other end. It also greatly increases the resistance of V.32 to noise on the line. In the past, V.32 has been expensive to implement because the technology it requires is complex. Because a one-way, 9600-bps stream uses almost the entire bandwidth of the phone line, V.32 modems implement *echo cancellation*, meaning that they cancel out the overlapping signal that their own modems transmit and just listen to the other modem's signal. This procedure is complicated and costly. Recent advances in lower-cost chipsets make these modems inexpensive, so they are becoming the *de facto* 9600-bps standard.

V.32bis. V.32bis is a relatively new 14,400-bps extension to V.32. This protocol uses TCQAM modulation at 2400 baud to transmit 6 bits per baud, for an effective rate of 14,400 bits per second. The trellis coding makes the connection more reliable. This protocol is also a full-duplex modulation protocol, with fallback to V.32 if the phone line is impaired. Although this high-speed standard is newly developed, it is rapidly becoming the communications standard for dial-up lines because of its excellent performance and resistance to noise. I recommend the V.32bis-type modem.

V.32fast. V.32fast is a new standard being proposed to the CCITT. V.32fast will be an extension to V.32 and V.32bis, but will offer a transmission speed of 28,800 bits per second. This standard, when approved, probably will be as advanced as modem communications ever gets. Looming on the horizon is that the phone system eventually will be digital. All further development on analog transmission schemes will end, and new digital modems will be developed. V.32fast will be the best and last of the analog protocols when it debuts.

Error-Correction Protocols. *Error correction* refers to a capability that some modems have to identify errors during a transmission, and to automatically resend data that appears to have been damaged in transit. For error correction to work, both modems must adhere to the same correction standard. Fortunately, most modem manufacturers follow the same error-correction standards.

V.42. V.42 is an error-correction protocol, with fallback to MNP 4. MNP stands for Microcom Networking Protocol (covered later in this section), and Version 4 is an error-correction protocol as well. Because the V.42 standard includes MNP compatibility through Class 4, all MNP 4 compatible modems can establish error-controlled connections with V.42 modems. This standard uses a protocol called LAPM (Link Access Procedure for Modems). LAPM, like MNP, copes with phone-line impairments by automatically retransmitting data corrupted during transmission, assuring that only error-free data passes through the modems. V.42 is considered to be better than MNP 4 because it offers about a 20 percent higher transfer rate due to more intelligent algorithms.

Data-Compression Standards. Data compression refers to a built-in capability in some modems to compress the data they're sending, thus saving time and money for long-distance modem users. Depending on the type of files that are sent, data can be compressed to 50 percent of its original size, effectively doubling the speed of the modem.

V.42bis. V.42bis is a CCITT data-compression standard similar to MNP Class 5, but providing about 35 percent better compression. V.42bis is not actually compatible with MNP Class 5, but nearly all V.42bis modems include the MNP 5 data-compression capability as well.

This protocol can sometimes quadruple throughput, depending on the compression technique used. This fact has led to some mildly false advertising: for example, a 2400-bps V.42bis modem might advertise "9600 bps throughput" by including V.42bis as well, but this would be possible in only extremely optimistic cases, such as in sending text files that are very loosely packed. In the same manner, many 9600-bps V.42bis makers now advertise "up to 38.4K bps throughput" by virtue of the compression. Just make sure that you see the truth behind such claims.

V.42bis is superior to MNP 5 because it analyzes the data first, and then determines whether compression would be useful. V.42bis only compresses data that needs compression. Files found on bulletin board systems often are compressed already (using ARC, PKZIP, and similar programs). Further attempts at compressing already compressed data can increase the size of the data and slow things down. MNP 5 always attempts to compress the data, which slows down throughput on previously compressed files. V.42bis, however, compresses only what benefits from the compression.

To negotiate a standard connection using V.42bis, V.42 also must be present. Therefore, a modem with V.42bis data compression is assumed to include V.42 error correction. These two protocols combined result in an error-free connection that has the maximum data compression possible.

Proprietary Standards. In addition to the industry-standard protocols for modulation, error correction, and data compression that generally are set forth or approved by the CCITT, several protocols in these areas were invented by various companies and included in their products without any official endorsement by the CCITT. Some of these protocols are quite popular and have become pseudo-standards of their own.

The most successful proprietary protocols are the MNP (Microcom Networking Protocols) that were developed by Microcom. These error-correction and data-compression protocols are supported widely by other modem manufacturers as well. Another company successful in establishing proprietary protocols as limited standards is U.S. Robotics, with its HST (high speed technology) modulation protocols. Because of an aggressive marketing campaign with bulletin board system operators, it captured a large portion of the market with its products.

This section examines these and other proprietary modem protocols.

HST. The HST is a 14400-bps and 9600-bps modified half-duplex proprietary modulation protocol used by U.S. Robotics. Although common in bulletin board systems, the HST probably is destined for extinction within the next few years as V.32 modems become more competitive in price. HST modems run at 9600 bps or 14400 bps in one direction, and 300 or 450 bps in the other direction. This is an ideal protocol for interactive sessions. Because echo-cancellation circuitry is not required, costs are lower.

U.S. Robotics also makes modems that use standard protocols as well as *dual standards*—modems that incorporate both V.32bis and HST protocols. This gives you the best of the standard and proprietary worlds and enables you to connect to virtually any other system at that system's maximum communications rate. I use and recommend the dual-standard modems.

DIS. The DIS is a 9600-bps proprietary modulation protocol by CompuCom, which uses dynamic impedance stabilization (DIS), with claimed superiority in noise rejection over V.32. Implementation appears to be very inexpensive, but like HST, only one company makes modems with the DIS standard. Because of the lower costs of V.32 and V.32bis, this proprietary standard will likely disappear.

MNP. MNP (Microcom Networking Protocol) offers end-to-end error correction, meaning that the modems are capable of detecting transmission errors and requesting retransmission of corrupted data. Some levels of MNP also provide data compression.

As MNP evolved, different classes of the standard were defined, describing the extent to which a given MNP implementation supports the protocol. Most current implementations support Classes 1 through 5. Higher classes usually are unique to modems manufactured by Microcom, Inc. because they are proprietary.

MNP generally is used for its error-correction capabilities, but MNP Classes 4 and 5 also provide performance increases, with Class 5 offering real-time data compression. The lower classes of MNP usually are not important to you as a modem user, but they are included for completeness in the following list:

- MNP Class 1 (block mode) uses asynchronous, byte-oriented, half-duplex (one-way) transmission. This method provides about 70 percent efficiency and error correction only, so it's rarely used today.
- MNP Class 2 (stream mode) uses asynchronous, byte-oriented, full-duplex (two-way) transmission. This class also provides error correction only. Because of *protocol overhead* (the time it takes to establish the protocol and operate it), throughput at Class 2 is only about 84 percent of that for a connection without MNP, delivering about 202 cps (characters per second) at 2400 bps (240 cps is the theoretical maximum). Class 2 is used rarely today.
- MNP Class 3 incorporates Class 2 and is more efficient. It uses a synchronous, bit-oriented, full-duplex method. The improved procedure yields throughput about 108 percent of that of a modem without MNP, delivering about 254 cps at 2400 bps.
- MNP Class 4 is a performance-enhancement class that uses Adaptive Packet Assembly and Optimized Data Phase techniques. Class 4 improves throughput and performance by about 5 percent, although actual increases depend on the type of call and connection, and can be as high as 25 percent to 50 percent.
- MNP Class 5 is a data-compression protocol that uses a real-time adaptive algorithm. It can increase throughput up to 50 percent, but the actual performance of Class 5 depends on the type of data being sent. Raw text files allow the highest increase, although program files cannot be compressed as much and the increase is smaller. On precompressed data (files already compressed with ARC, PKZIP, and so on), MNP 5 *decreases* performance, and therefore is often disabled on BBS systems.

V-Series. The Hayes V-series is a proprietary error-correction protocol by Hayes that was used in some of its modems. Since the advent of lower cost V.32 and V.32bis modems (even from Hayes), the V-series has all but become extinct. These modems used a modified V.29 protocol, which is sometimes called a *ping-pong protocol* because it has one high-speed channel and one low-speed channel that alternate back and forth.

CSP. The CSP (CompuCom Speed Protocol) is an error-correction and data-compression protocol available on CompuCom DIS modems.

FAXModem Standards. Facsimile technology is a science unto itself, although it has many similarities to data communications. These similarities have led to the combination of data and faxes into the same modem. You now can purchase a single board that will send and receive both data and faxes; all the major modem manufacturers have models that support this capability.

Over the years, the CCITT has set international standards for fax transmission. This has led to the grouping of faxes into one of four groups. Each group (I through IV) uses different technology and standards for transmitting and receiving faxes. Groups I and II are relatively slow and provide results that are unacceptable by today's standards. Group III is the standard in use today by virtually all fax machines, including those combined with modems. Whereas Groups I through III are analog in nature (similar to modems), Group IV is digital and designed for use with ISDN or other digital networks. Because the telephone system has not converted to a digital system yet, there are very few Group IV fax systems available.

If you are interested in detailed information on the technical fax specifications, you can contact Telecommunications Industry Association or Global Engineering Documents.

Group III Fax. There are two general subdivisions within the Group III fax standard—Class 1 and Class 2. Many times you will hear about a FAXModem supporting Group III, Class 1 fax communications. This simply indicates the protocols that the board is able to send and receive. If your FAXModem does this, it can communicate with most of the other fax machines in the world. In FAXModems, the Class 1 specification is implemented by an additional group of modem commands that the modem translates and acts upon.

Earlier you learned about the V.29 modulation standard. As stated in that section, this standard is used for Group III fax transmissions.

Modem Recommendations. Today the cost of 9600-bps modems has dropped to just over $100 for one of these modems. You can even find 14400-bps modems for under $200.

Today, most modems come with multiple forms of error correction or data compression. Based on the discussions earlier in this chapter, you should search for the modem that offers the best combination of speed, error correction, and data compression. Probably the most universal modem is the U.S. Robotics Courier HST Dual Standard—a high-speed modem that uses both the industry standard ultra-high speed V.32bis protocol and U.S. Robotics' own transmission standard as well. This modem also includes V.42bis, which enables throughput to hit 38.4K bps with the right data. The multiple protocols and standards in this modem make possible connections to almost any other modem at its full capability. You are limited only by the speed and protocols of the modem you are calling. This device's only drawback is its price (which has been coming down lately), but its flexibility makes it cost effective.

Secrets of Modem Negotiation. If you are curious about the complex negotiations that occur when two modems connect, two detailed descriptions of *modem handshaking* follow. The two examples of connections use V.22bis and V.32. These sequences may differ slightly depending on your modem, and can get more complicated when you combine different modem types into one box. Making a V.22bis connection between two modems involves the following sequence of events:

1. The answering modem detects a ring, goes off-hook, and waits for at least two seconds of *billing delay* (required by phone company rules so that no data passes before the network recognizes that the call has been connected).
2. The answering modem transmits an *answer tone* (described in CCITT Recommendation V.25, occurs at 2100 Hz, and lasts 3.3 ± 0.7 seconds). The answer tone tells manual-dial originators that they have reached a modem and can put their calling modem in data mode, and informs the network that data is going to be transferred so that echo suppressors in the network can be disabled. If the echo suppressors remain enabled, you cannot transmit in both directions at the same time. (The originating modem remains silent throughout this period.)
3. The answering modem goes silent for 75 ± 20 milliseconds (ms) to separate the answer tone from the signals that follow.
4. The answering modem transmits unscrambled binary 1s at 1200 bits per second (USB1), which cause the static, or harsh sound, you hear after the answer tone. This sound is slightly higher in pitch than the answer tone because the signal's major components are at 2250 Hz and 2550 Hz.
5. The originating modem detects the USB1 signal in 155 ± 10 ms, and remains silent for 456 ± 10 ms.
6. The originating modem transmits unscrambled double-digit 00s and 11s at 1200 bits (S1) for 100 ± 3 ms. A Bell 212 or V.22 modem does not transmit this S1 signal, and it is by the presence or absence of this single 100-ms signal that V.22bis knows whether to fall back to 1200-bps operation.
7. When the answering modem (which is still transmitting the USB1 signal) detects the S1 signal from the originator, it also sends 100 ms of S1 so that the originating modem knows that the answerer is capable of 2400-bps operation.
8. The originating modem then switches to sending scrambled binary 1s at 1200 bits (SB1). *Scrambling* has nothing to do with encryption or security, but is simply a method by which the signal is *whitened*, or randomized, to even out the power across the entire bandwidth. *White noise* is a term given by engineers to totally random noise patterns.
9. The answering modem switches to sending SB1 for 500 ms.
10. The answering modem switches to sending scrambled 1s at 2400 bps for 200 ms. After that, it is ready to pass data.
11. 600 ms after the originating modem hears SB1 from the answerer, it switches to sending scrambled 1s at 2400 bps. It does this for 200 ms, and then is ready to pass data.

The signals involved in a V.32 connection are more complicated than V.22bis because of the need to measure the total round-trip delay in the circuit so that the echo cancellers work. Making a V.32 connection involves the following sequence of events:

1. The answering modem detects a ring, goes off-hook, and waits two seconds (the billing delay).
2. The answering modem transmits a V.25 answer tone, but it is different from the preceding example. The phase of signal is reversed every 450 ms, which sounds like little clicks in the signal. These phase reversals inform the network that the modems themselves are going to do echo cancellation, and that any echo cancellers in the network should be disabled so as not to interfere with the modems.
3. The originating V.32 modem does not wait for the end of the answer tone. After one second, it responds with an 1800-Hz tone, which in V.32 is known as signal AA. Sending this signal before the end of the answer tone enables the answering modem to know, very early, that it is talking to another V.32 modem.
4. After the answer tone ends (3.3 ± 0.7 seconds), if the answering modem heard signal AA, it proceeds to try to connect as V.32 immediately. If it did not hear AA, it first tries, for three seconds, to connect as a V.22bis modem (sends signal USB1 and waits for a response). If it does not get a response to USB1, it goes back to trying to connect as a V.32 modem because of the possibility that the calling V.32 modem didn't hear the answer tone, was manually dialed and switched to data mode late, or is an older V.32 model that does not respond to the answer tone.
5. To connect in V.32, the answering modem sends signal AC, which is 600 Hz and 3000 Hz sent together, for at least 64 *symbol intervals* (1/2400 of a second). It then reverses the phase of the signal, making it into signal CA.
6. When the originating modem detects this phase reversal, in 64 ± 2 symbol intervals, it reverses the phase of its own signal, making AA into CC.
7. When the answer modem detects this phase reversal (in 64 ± 2 symbol intervals), it again reverses the phase of its signal, making CA back into AC. This exchange of phase reversals enables the modems to accurately time the total propagation (round trip) delay of the circuit so that the echo cancellers can be set to properly cancel signal echoes.
8. The modems go into a half-duplex exchange of training signals to train the adaptive equalizers, test the quality of the phone line, and agree on the data rate to be used. The answering modem transmits first, from 650 ms to 3525 ms, and then goes silent.
9. The originating modem responds with a similar signal, but then leaves its signal on, while the answering modem responds one more time, establishing the final agreed-on data rate.
10. Both modems then switch to sending scrambled binary 1 (marks) for at least 128 symbol intervals, and then are ready to pass data.

As you can see, these procedures are quite complicated. Although you do not need to understand these communications protocols to use a modem, you can get an idea of what you're hearing when the connection is being established.

Parallel Ports

A parallel port has eight lines for sending all the bits for one byte of data, simultaneously, across eight wires. This interface is fast and usually is reserved for printers rather than computer-to-computer communications. The only problem with parallel ports is that cables cannot be extended for any great length without amplifying the signal, or errors occur in the data. Table 11.7 shows the pinout for a standard PC parallel port.

Table 11.7 25-Pin PC-Compatible Parallel Port Connector

Pin	Description	I/O
1	–Strobe	Out
2	+Data Bit 0	Out
3	+Data Bit 1	Out
4	+Data Bit 2	Out
5	+Data Bit 3	Out
6	+Data Bit 4	Out
7	+Data Bit 5	Out
8	+Data Bit 6	Out
9	+Data Bit 7	Out
10	–Acknowledge	In
11	+Busy	In
12	+Paper End	In
13	+Select	In
14	–Auto Feed	Out
15	–Error	In
16	–Initialize Printer	Out
17	–Select Input	Out
18	–Data Bit 0 Return (GND)	In
19	–Data Bit 1 Return (GND)	In
20	–Data Bit 2 Return (GND)	In
21	–Data Bit 3 Return (GND)	In
22	–Data Bit 4 Return (GND)	In
23	–Data Bit 5 Return (GND)	In
24	–Data Bit 6 Return (GND)	In
25	–Data Bit 7 Return (GND)	In

Over the years, four types of parallel ports have evolved: Original, Type 1, Type 3, and Enhanced Parallel Port. The following sections discuss each of these types.

Original Unidirectional. The original IBM PC did not have different types of parallel ports available. The only port available was the parallel port used to send information from the computer to a device, such as a printer. This is not to say that bidirectional parallel ports were not available; indeed, they were common in other computers on the market and with hobbyist computers at the time.

The unidirectional nature of the original PC parallel port is consistent with its primary use—that is, sending data to a printer. There were times, however, when it was desirable to have a bidirectional port—for example, when you need feedback from a printer, which is common with PostScript printers. This could not be done with the original unidirectional ports.

Type 1 Bidirectional. With the introduction of the PS/2 in 1987, IBM introduced a bidirectional parallel port. This opened the way for true communications between the computer and the peripheral across the parallel port. This was done by defining a few of the previously unused pins in the parallel connector, and defining a status bit to indicate in which direction information was traveling across the channel.

In IBM documentation, this original PS/2 port became known as a Type 1 parallel port. Other vendors also introduced third-party ports that were compatible with the Type 1 port; they were based on chips from Chips & Technologies, Inc. Unless you specifically configure the port for bidirectional use, however, the PS/2 Type 1 port functions the same as the original unidirectional port in the PC. This configuration is done with the configuration disk that accompanies the PS/2.

Type 3 DMA. With the introduction of the PS/2 Models 57, 90, and 95, IBM introduced the Type 3 parallel port. This port featured greater throughput by use of direct memory access (DMA) techniques. You may be wondering why IBM skipped from Type 1 to Type 3. In reality, they did not. There is a Type 2 parallel port, and it served as a predecessor to the Type 3. It is only slightly less capable, but was never used widely in any IBM systems.

The types of parallel ports described earlier all use the CPU to control operations. In these ports, the CPU sends one byte to the I/O addresses used by the port, tests to see whether it was sent, and then sends the next. This process continues until all the information to be sent has been processed. The Type 3 port, however, enables you to define a block of memory that you want transmitted (perhaps a file stored in memory), and then enables the port controller to pace the sending of the information. This frees the CPU to perform other tasks, thereby increasing throughput.

Enhanced Parallel Port (EPP) Specification. This is a newer specification from Intel, sometimes referred to as the *Fast Mode parallel port.* While the IBM Type 3 port is designed to enable you to move large amounts of data without tying up the CPU, the Fast Mode port is aimed at quick bidirectional communication with intelligent peripherals. To do this, a Fast Mode port, when operating in that mode, redefines the parallel pins so they actually become bus control lines to augment communication and control between your system and the external device.

Parallel Port Configuration

Parallel-port configuration is not as complicated as it is for serial ports. Even the original IBM PC has BIOS support for three LPT ports, and DOS has always had this support as well. Table 11.8 shows the standard I/O address and interrupt settings for parallel port use.

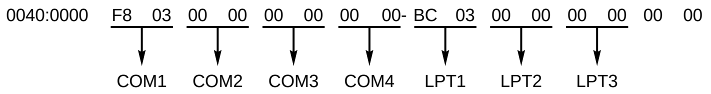
Table 11.8 Parallel Interface I/O Port Addresses and Interrupts

	LPTx		**I/O**	
System	**Std.**	**Alt.**	**Port**	**IRQ**
8-bit ISA		LPT1	3BCh	IRQ7
8-bit ISA	LPT1	LPT2	378h	None
8-bit ISA	LPT2	LPT3	278h	None
16-bit ISA		LPT1	3BCh	IRQ7
16-bit ISA	LPT1	LPT2	378h	IRQ5
16-bit ISA	LPT2	LPT3	278h	None
	LPTx		**I/O**	
System	**Std.**	**Alt.**	**Port**	**IRQ**
All MCA	LPT1		3BCh	IRQ7
All MCA	LPT2		378h	IRQ7
All MCA	LPT3		278h	IRQ7

Because the BIOS and DOS always have provided three definitions for parallel ports, problems with older systems are infrequent. Problems can arise, however, from the lack of available interrupt-driven ports for the ISA bus systems. Normally, an interrupt-driven port is not absolutely required for printing operations; in fact, many programs do not use the interrupt-driven capability. Many programs do use the interrupt, however, such as network print programs and other types of background or spooler-type printer programs. Also, any high-speed, laser-printer utility programs often would use the interrupt capabilities to allow for printing. If you use these types of applications on a port that is not interrupt driven, you see the printing slow to a crawl, if it works at all. The only solution is to use an interrupt-driven port. Note that because MCA bus PS/2 systems can share interrupts, they are completely exempt from this type of problem, and all parallel ports in these systems are interrupt driven on IRQ7.

To configure parallel ports in ISA bus systems, you probably will have to set jumpers and switches. Because each board on the market is different, you always should consult the OEM manual for that particular card if you need to know how the card should or can be configured. IBM includes this information with each card's documentation. IBM also offers technical-reference options and adapters manuals as well as hardware-maintenance and service manuals, which also cover in detail all the jumper and switch settings for IBM adapter cards. Other manufacturers simply include with the card a manual that describes the card and includes the configuration information. The PS/2 MCA bus systems have an automatic or software-driven configuration that eliminates these configuration hassles.

Parallel Port Devices. As already mentioned, the original IBM PC envisioned that the parallel port would be used only for communicating with a printer. Over the years, the number of devices that can be used with a parallel port has increased tremendously. You now can find everything from tape backup units to LAN adapters to CD-ROMs that connect through your parallel port.



Perhaps one of the most common uses for bidirectional parallel ports is to transfer data between your system and another, such as a laptop computer. If both systems utilize a Type 3 port, you can actually communicate at rates of up to 2M per second, which rivals the speed of some hard disk drives. This capability has led to an increase in software to serve this niche of the market. If you are interested in such software (and the parallel ports necessary to facilitate the software), you should refer to the reviews that periodically appear in sources such as *PC Magazine*.

Detecting Serial and Parallel Ports with DEBUG

If you cannot tell which ports (parallel and serial) the computer is using, check for the I/O ports by using DEBUG.

To use DEBUG, follow these steps:

1. Run DEBUG.
2. At the DEBUG prompt, type **D 40:0** and press Enter. This step displays the hexadecimal values of the active I/O port addresses—first serial and then parallel. Figure 11.3 shows a sample address.

Fig. 11.3
DEBUG used to display installed serial and parallel port I/0 port addresses.

   The address for each port is shown in the corresponding position. Because addresses are stored as words, the byte values are swapped and should be read backward.
   This example indicates one serial port installed at 03F8 and one parallel port installed at 03BC.

3. To exit DEBUG, press Q and press Enter.

Testing Serial Ports

You can perform several tests on serial and parallel ports. The two most common types of tests involve software only, or both hardware and software. The software-only tests are done with diagnostic programs such as Microsoft's MSD, while the hardware and software tests involve using a wrap plug to perform loopback testing.

Microsoft Diagnostics (MSD). MSD is a diagnostic program supplied with MS-DOS 6 or Microsoft Windows. Early versions of the program also were shipped with some Microsoft applications such as Microsoft Word for DOS.

To use MSD, switch to the directory in which it is located. This is not necessary, of course, if the directory containing the program is in your search path—which is often the case with the DOS 6- or Windows-provided versions of MSD. Then simply type **MSD** at the DOS prompt and press Enter. Soon you see the MSD screen.

Choose the Serial Ports option by pressing S. Notice that you are provided information about what type of serial chip you have in your system, as well as information about what ports are available. If any of the ports are in use (for example, a mouse), that information is provided as well.

MSD is helpful in at least determining whether your serial ports are responding. If MSD cannot determine the existence of a port, it does not provide the report indicating that the port exists. This sort of "look and see" test is the first action I usually take to determine why a port is not responding.

Advanced Diagnostics Using Loopback Testing. One of the most useful tests is the *loopback* test, which can be used to ensure the correct function of the serial port, as well as any attached cables. Loopback tests basically are internal (digital), or external (analog). Internal tests can be run simply by unplugging any cables from the port and executing the test via a diagnostics program.

The external loopback test is more effective. This test requires that a special loopback connector or wrap plug be attached to the port in question. When the test is run, the port is used to send data out to the loopback plug, which simply routes the data back into the port's receive pins so that the port is transmitting and receiving at the same time. A loopback or wrap plug is nothing more than a cable doubled back on itself. Most diagnostics programs that run this type of test include the loopback plug, and if not, these types of plugs can be purchased easily or even built. See Appendix A, "PC Technical Reference Section," for the necessary diagrams to construct your own wrap plugs.

If you want to purchase a wrap plug, I recommend the IBM tri-connector wrap plug. IBM sells this triple plug, as well as individual wrap plugs, under the following part numbers:

Description	IBM Part Number
Parallel-port wrap plug, 25-pin	8529280
Serial-port wrap plug, 9-pin (AT)	8286126
Tri-connector wrap plug	72X8546

As for the diagnostics software, IBM's own Advanced Diagnostics can be used to test serial ports. If you have a PS/2 system with Micro Channel Architecture, IBM already has given you the Advanced Diagnostics on the Reference Disk that came with the system. To activate this normally hidden Advanced Diagnostics, press Ctrl-A at the Reference Disk's main menu. For IBM systems that are not MCA, the Advanced Diagnostics must be purchased.

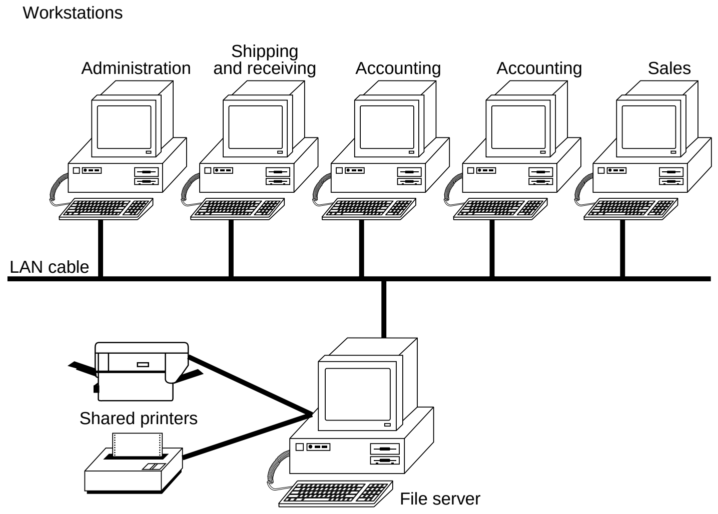
For any system, you can use the serial port tests in the comprehensive diagnostics packages sold by several companies as a replacement for the IBM- (or other manufacturer) supplied Advanced Diagnostics. Programs such as Micro-Scope from Micro 2000, Service Diagnostics from Landmark, or QA-Plus FE from Diagsoft all have this type of test as part of their package. All include the necessary three-wrap plugs as well. See Appendix B, "Vendor List," or Chapter 13, "Floppy Disk Drives and Controllers," for more information on these programs.

For testing serial ports, as well as any modems that are attached, I highly recommend an inexpensive ($19.95) program called the Modem Doctor. This comprehensive serial port and modem test program enables you to go beyond the simple loopback tests and test the complete communications system, including the cable and modem. The program is especially useful with the U.S. Robotics modems or any other modem with a Hayes-compatible command structure. The program takes command of the modem and runs a variety of tests to determine whether it is functioning correctly.

Testing Parallel Ports

Testing parallel ports is, in most cases, simpler than testing serial ports. The procedures you use are effectively the same as those used for serial ports, except that when you use the diagnostics software, you choose the obvious choices for parallel ports rather than serial ports.

Not only are the software tests similar, but the hardware tests require the proper plugs for the loopback tests on the parallel port. You can use the tri-connector wrap plug recommended earlier, or you can purchase an individual parallel port wrap plug. If you want the individual plug, ask for IBM part number 8529228.



Diagnosing Problems with Serial and Parallel Ports

To diagnose problems with serial and parallel ports, you need diagnostics software and a wrap plug for each type of port. The diagnostics software works with the wrap plugs to send signals through the port. The plug wraps around the port so that the same port receives the information it sent. This information is verified to ensure that the port works properly. For further information, see Chapter 14, “Hard Disk Drives and Controllers.”

You next turn your attention to local area network components and concepts.

Understanding the Components of a LAN

A local area network (LAN) enables you to share files, share applications, use multiuser software products, share printers, use electronic mail, share disk space, share modems, and otherwise make a collection of computers work as a team.

A LAN is a combination of computers, LAN cables, network adapter cards, network operating system software, and LAN application software. (You sometimes see *network operating system* abbreviated as *NOS*.) On a LAN, each personal computer is called a *workstation*, except for one or more computers designated as *file servers*. Each workstation and file server contains a network adapter card. LAN cables connect all the workstations and file servers. In addition to its local operating system (usually DOS), each workstation runs network software that enables the workstation to communicate with the file servers. In turn, the file servers run network software that communicates with the workstations and serves up files to those workstations. LAN-aware application software runs at each workstation, communicating with the file server when it needs to read and write files. Figure 11.4 illustrates the components that make up a LAN.

Fig. 11.4

The components of a LAN.

Workstations

A LAN is made up of computers. You will find two kinds of computers on a LAN: the workstations, usually manned by people; and the file servers, usually located in a separate room or closet. The workstation works only for the person sitting in front of it, whereas a file server enables many people to share its resources. Workstations usually are intermediate-speed AT-class machines with an 80286 or better CPU, and they may have 1M to 4M of RAM. Workstations often have good-quality color or gray-scale VGA monitors, as well as high-quality keyboards, but these are characteristics that make them easy to use and are not required to make the LAN work. A workstation also usually has an inexpensive, slow, small hard disk.

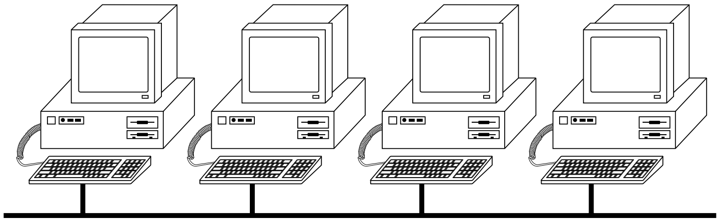
Some workstations, called *diskless workstations*, do not have a disk drive of their own. Such workstations rely completely on the LAN for their file access.

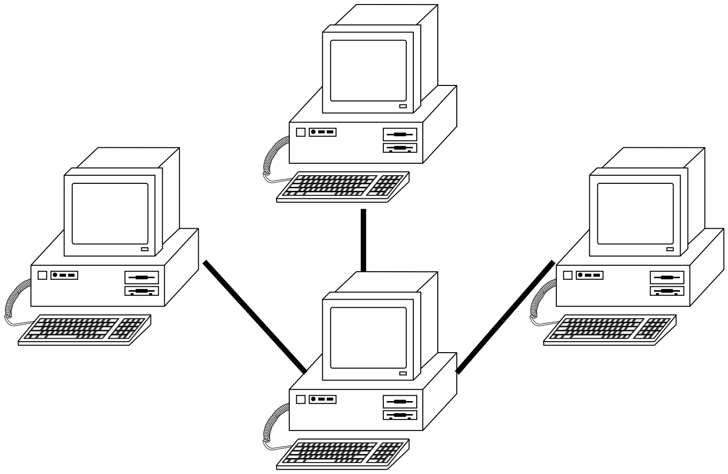
File Servers

In contrast to the workstations, a *file server* is a computer that serves all the workstations—primarily storing and retrieving data from files shared on its disks. File servers usually are fast 386-, 486-, or Pentium-based computers, running at 25 MHz or faster and with 8M or more of RAM. File servers usually have only monochrome monitors and inexpensive keyboards, because people do not interactively use file servers. The file server normally operates unattended. A file server almost always has one or more fast, expensive, large hard disks, however.

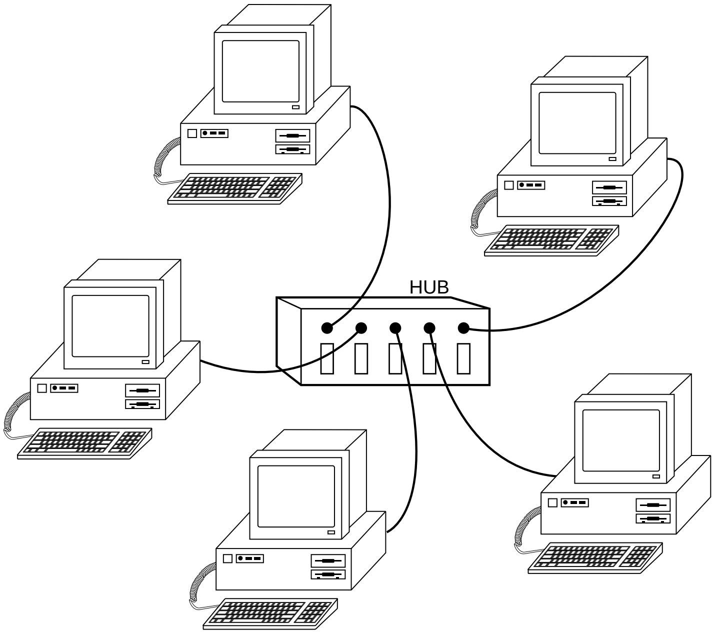
Servers must be high-quality, heavy-duty machines because, in serving the whole network, they do many times the work of an ordinary workstation computer. In particular, the file server's hard disk(s) need to be durable and reliable.

You most often will see a computer dedicated to the task of being a file server. Sometimes, on smaller LANs, the file server doubles as a workstation. Serving an entire network is a big job that does not leave much spare horsepower to handle workstation duties, however, and if an end user locks up the workstation that serves as the file server, your network also locks up.





Under a heavy load, if there are 20 workstations and one server, each workstation can use only one-twentieth of the server's resources. In practice, though, most workstations are idle most of the time—at least from a disk-file-access point of view. As long as no other workstation is using the server, your workstation can use 100 percent of the server's resources.



LAN Cables

LAN cable comes in different varieties. You may use thin coaxial wire (referred to as *Thinnet* or *CheaperNet*) or thick coaxial wire (*ThickNet*). You may use shielded twisted pair (*STP*), which looks like the wire that carries electricity inside the walls of your house, or unshielded twisted pair (*UTP*), which looks like telephone wire. You may even use fiber optic cable. Fiber optic cable works more over longer distances than other types of cable, at faster speeds. Fiber optic cable installation and fiber-optic-based network adapters can be expensive, however. The kind of wire you use depends mostly on the kind of network adapter cards you choose. The next section discusses network adapter cards.

Each workstation is connected with cable to the other workstations and to the file server. Sometimes a single piece of cable wends from station to station, visiting all the servers and workstations along the way. This cabling arrangement is called a *bus* or *daisy-chain topology*, as shown in figure 11.5. (A *topology* is simply a description of the way the workstations and servers are physically connected.)

Sometimes separate cables run from a central place, such as a file server, to each workstation. Figure 11.6 shows this arrangement, called a *star*. Sometimes the cables branch out repeatedly from a root location, forming the *star-wired tree* shown in figure 11.7. Daisy-chained cabling schemes use the least cable but are the hardest to diagnose or bypass when problems occur.

Fig. 11.5
The linear bus topology, attaching all network devices to a common cable.

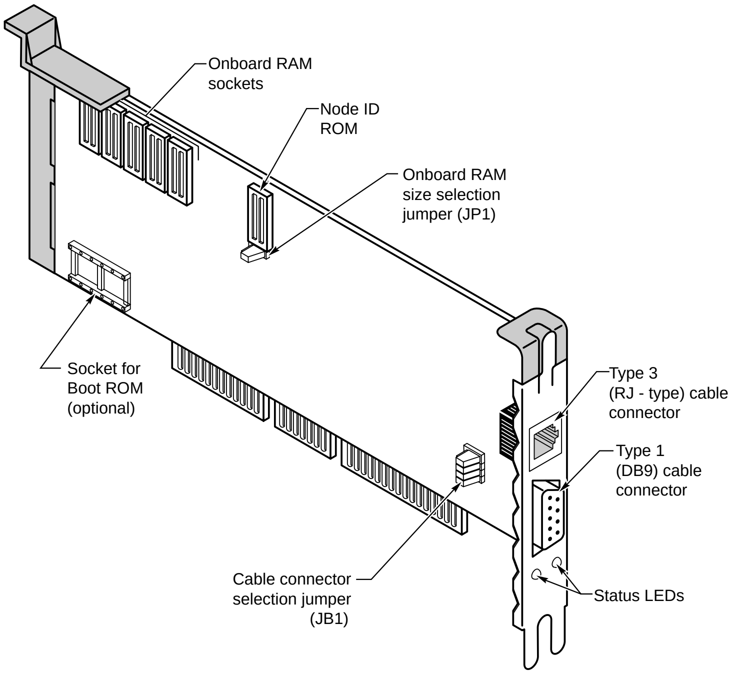
Fig. 11.6
The star topology, connecting the LAN's computers and devices with cables that radiate outward, usually from a file server.

Fig. 11.7
The star-wired tree topology, linking the LAN's computers and devices to one or more central hubs, or access units.

If you have to run cables through walls or ceilings, installing the cable can be the most expensive part of setting up a LAN. At every branching point, special fittings connect the intersecting wires. Sometimes you also need various black boxes such as hubs, repeaters, or access units.

A few companies, such as Motorola, are pioneering a type of LAN that does not require cables at all. Such a *wireless LAN* uses infrared or radio waves to carry network signals from computer to computer.

Planning the cabling layout, cutting the cable, and installing the cables and fittings are jobs usually best left to experienced workers. If the fittings are not perfect, you may get electronic echoes on the network, which cause transmission errors. Coaxial cable costs about 15 cents per foot, whereas shielded twisted pair probably costs more than $1 per



foot. This sounds like a big expense for a large LAN, but the cost of installing cable, at about $45 per hour, overshadows the cost of the cable itself. The only time you might consider installing LAN cable yourself is when you have a group of computers located on adjacent desks and you do not have to enter the walls or ceiling with the cable.

Building codes almost always require you to use fire-proof *plenum* cables. Chapter 5, "Bus Slots and Specifications," explains LAN cables in detail. For now, you should know that plenum cables are more fire-resistant than some other cables. You would be very upset if you installed ordinary cable and were later told by the building inspector to rip out the cable and start over with the proper kind.

Network Adapters

A network adapter card, like a video display adapter card, fits in a slot in each workstation and file server. Your workstation sends requests through the network adapter to the file server. The workstation receives responses through the network adapter when the file server delivers all or a portion of a file to that workstation. The sending of these requests and responses is the LAN's equivalent of reading and writing files on your PC's local hard disk. If you're like most people, you probably think of reading and writing files in terms of loading or saving your work.

Only two network adapters can communicate with each other at the same time on a LAN. This means that other workstations have to wait their turn if one person's workstation currently is accessing the file server (processing the requests and responses that deliver a file to the workstation). Fortunately, such delays are usually not noticeable. The LAN gives the appearance of many workstations accessing the file server simultaneously.

LANtastic adapters have two connectors on the back to attach the incoming and outgoing cables. Ethernet connectors have a single T connector, a D-shaped 15-pin connector, a connector that looks like a telephone jack, or sometimes a combination of all three. Token Ring adapters have a 9-pin connector and sometimes a telephone jack outlet. Figure 11.8 shows a high-performance Token Ring adapter with both kinds of connectors.

Fig. 11.8
The Thomas-Conrad 16/4 Token Ring adapter (with a 9-pin connector and a telephone wire connector).

Cards with two or more connectors enable you to choose from a wider variety of LAN cables. A Token Ring card with two connectors, for example, enables you to use shielded twisted pair (STP) or unshielded twisted pair (UTP, or telephone wire) cable.

The LAN adapter card listens to all the traffic going by on the cable, and filters out just the messages destined for your workstation. The adapter hands these messages over to your workstation when the workstation is ready to attend to the messages. When the workstation wants to send a request to a server, the adapter card waits for a break in the cable traffic and inserts your message into the stream. The workstation also verifies that the message arrived intact, and resends the message if it arrived garbled.

Adapters range in price from less than $100 to much more than $1,000. What do you get for your money? Primarily, speed. The faster adapters can push data faster onto the cable, which means that the file server gets a request more quickly and sends back a response more quickly.

Data-Transfer Speeds on a LAN

Electrical engineers and technical people measure the speed of a network in *megabits per second*. Because a byte of information consists of 8 bits, you can divide the megabits per second rating by 8 to find out how many millions of characters (bytes) per second the network can handle theoretically. Suppose that you want to transfer an entire 3 1/2-inch 720K floppy disk's worth of information across a LAN. The rated speed of the LAN is 4 megabits per second. Dividing 4 mbps by 8 tells you that the LAN theoretically can transmit 500 kilobytes (500K) of data per second. This is equivalent to an average hard disk's transfer rate. The data from the 720K floppy disk takes at least a few seconds to transfer, as you can see from these rough calculations.

In practice, a LAN is slower than its rated speed. In fact, a LAN is no faster than its slowest component. If you were to transfer 720K of data from one workstation's hard disk to the file server, the elapsed time would include not only the transmission time but also the workstation hard disk retrieval time, the workstation processing time, and the file server's hard disk and server CPU processing times. The transfer rate of your hard disk, which in this case is probably the slowest component involved in the copying of the data to the server, governs the rate at which data flows to the file server. Other people's requests interleave with your requests on the LAN, and the total transfer time may be longer because the other people are using the LAN at the same time you are.

If you transfer the data from a 720K floppy disk to the file server, you see that it takes even longer. Floppy disk drives, as you know, are slower than hard disks. Your workstation uses the network in small bursts as it reads the data from the floppy disk. The workstation cannot send data across the LAN, in this case, any faster than it can read the data from the disk.

LANtastic Adapters. Artisoft makes both Ethernet and its own proprietary network adapter cards. Artisoft's proprietary model is called a LANtastic adapter, which is a little confusing because Artisoft also makes a network operating system called LANtastic. The LANtastic adapter operates at a rate of 2 mbps, and it uses four-conductor cable strung out in a snaking path that connects to all the workstations. Installation is easy if you do not have to put the cable inside walls or the ceiling.

Ethernet and Token Ring are industry standards, while the LANtastic adapter has a proprietary design. Most people choose Ethernet or Token Ring network adapters when building a new LAN.

ARCnet Adapters. ARCnet is one of the oldest types of LAN hardware. It was originally a proprietary scheme of the Datapoint Corporation, but today many companies make ARCnet-compatible cards. ARCnet is a little slow, but it is forgiving of minor errors in installation. It is known for solid reliability, and ARCnet cable/adapter problems are easy to diagnose. ARCnet costs less than Ethernet. ARCnet operates something like Token Ring, but at the slower rate of 2.5 mbps. The section "Token Ring Adapters," later in this chapter, explains the basic principles on which ARCnet and Token Ring work.

Ethernet Adapters. Ethernet-based LANs enable you to interconnect a wide variety of equipment, including UNIX computers, Apple computers, IBM PCs, and IBM clones. You can buy Ethernet cards from dozens of competing manufacturers. Ethernet comes in three varieties (ThinNet, UTP, and ThickNet), depending on the thickness of the cabling you use. ThickNet cables can span a greater distance, but they are much more expensive. Ethernet operates at a rate of 10 mbps.

Between *data transfers* (requests and responses to and from the file server), Ethernet LANs remain quiet. After a workstation sends a request across the LAN cable, the cable falls silent again. What happens when two or more workstations (and/or file servers) attempt to use the LAN at the same time?

Suppose that one of the workstations wants to request something from the file server, just as the file server is sending a response to another workstation. A collision happens. (Remember that only two computers can communicate through the cable at a given moment.) Both computers—the file server and the workstation—back off and try again. Ethernet network adapters use something called *Carrier Sense, Multiple Access/Collision*

Detection (CSMA/CD) to detect the collision, and they each back off a random amount of time. This method effectively enables one computer to go first. With higher amounts of traffic, the frequency of collisions rises higher and higher, and response times become worse and worse. An Ethernet network actually can spend more time recovering from collisions than sending data. IBM and Texas Instruments, recognizing Ethernet's traffic limitations, designed Token Ring to solve the problem.

Token Ring Adapters. Except for fiber optic cables/adapters, Token Ring is the most expensive type of LAN. Token Ring uses shielded or unshielded twisted pair cable. Token Ring's cost is justified when you have a great deal of traffic from many workstations. You find Token Ring in large corporations with large LANs, especially if the LANs are attached to mainframe computers. Token Ring operates at a rate of 4 mbps or 16 mbps.

Workstations on a Token Ring LAN continuously pass an electronic token among themselves. The *token* is just a short message indicating that the network is idle. If a workstation has nothing to send, as soon as it receives the token, it passes the token on to the next downstream workstation. Only when a workstation receives the token can it send a message on the LAN. If the LAN is busy, and you want your workstation to send a message to another workstation or server, you must wait patiently for the token to come around. Only then can your workstation send its message. The message circulates through the workstations and file servers on the LAN, all the way back to you, the sender. The sender then sends a token to indicate that the network is idle again. During the circulation of the message, one of the workstations or file servers recognizes that the message is addressed to it and begins processing that message.

Token Ring is not as wasteful of LAN resources as this description makes it sound. The token takes almost no time at all to circulate through a LAN, even with 100 or 200 workstations. It is possible to assign priorities to certain workstations and file servers so that they get more frequent access to the LAN. And, of course, the token-passing scheme is much more tolerant of high traffic levels on the LAN than the collision-sensing Ethernet.

ARCnet and Token Ring are not compatible with one another, but ARCnet uses a similar token-passing scheme to control workstation and server access to the LAN.

Sometimes a station fumbles and "drops" the token. LAN stations watch each other and use a complex procedure to regenerate a lost token. Token Ring is quite a bit more complicated than Ethernet, and the LAN adapter cards are correspondingly more expensive.

Evaluating File Server Hardware

A typical file server consists of a personal computer that you dedicate to the task of sharing disk space, files, and a printer. On a larger network, you may instead use a personal computer especially built for file server work (a superserver), a minicomputer, or even a mainframe. No matter what sort of computer you choose to use as a server, the server and the workstations communicate with each other through network adapter cards and LAN cables.

A file server does many times the work of an ordinary workstation. You may type on the server's keyboard only a couple of times a day, and you may glance at the server's monitor only a few times. The server CPU and hard disk, however, take the brunt of responding to the file-service requests of all the workstations on the LAN.

If you consider your LAN an important investment in your office (it is hard to imagine otherwise), you will want to get the highest quality computer you can afford for the file server. The CPU should be an 80486 or Pentium chip and should be one of the faster models. The hard disk should be large and fast. But the most important consideration is that the CPU, the motherboard on which the CPU is mounted, and the hard disk should be rugged and reliable. Do not skimp on these components. *Downtime* (when the network is not operating) can be expensive because people cannot access their shared files to get their work done. Higher quality components will keep the LAN running without failure for longer periods of time.

In the same vein, you will want to set up a regular maintenance schedule for your file server. Over a few weeks, the fan at the back of the computer can move great volumes of air through the machine to keep it cool. The air may contain dust and dirt, which accumulates inside the computer. You should clean out the "dust bunnies" in the server every month or two. You do not replace components in the server as part of your regular preventive maintenance, but you will want to know whether a part is beginning to fail. You may want to acquire diagnostic software or hardware to periodically check the health of your file server. (Chapter 19, "Maintaining Your System: Preventive Maintenance, Backups, and Warranties," discusses the tools you can use to keep your file server fit and trim.)

The electricity the file server gets from the wall outlet may, from time to time, vary considerably in voltage (resulting in *sags* and *spikes*). To make your file server as reliable as possible, you should install an uninterruptible power supply between the electric company and your server. The UPS not only provides electricity in case of a power failure, but also conditions the line to protect the server from sags and spikes.

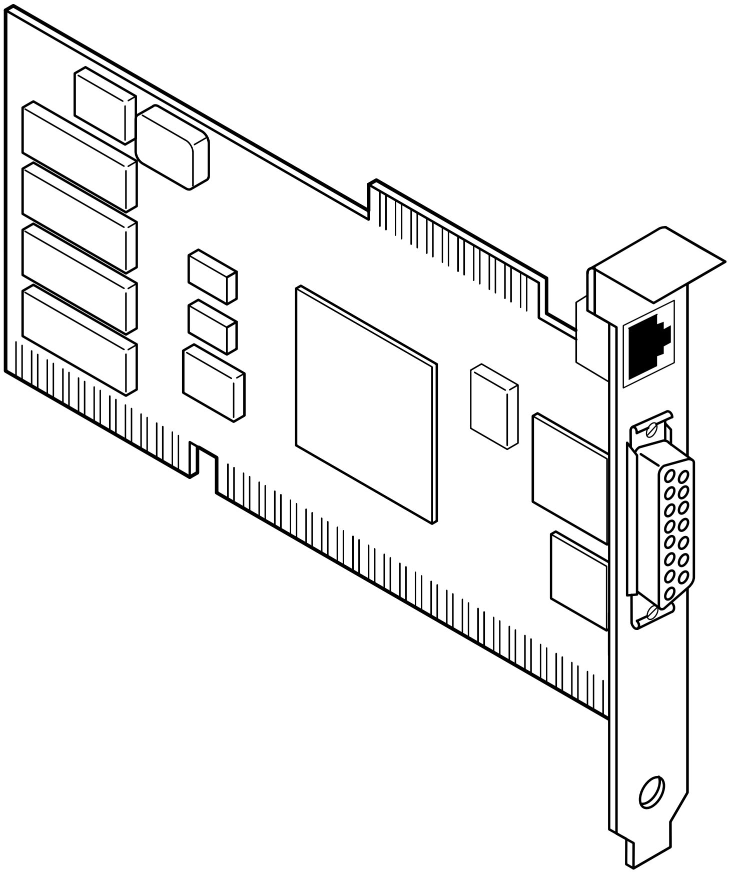
In general, you want to do whatever you can to make your network reliable, including placing the server away from public access areas.

Evaluating the File Server Hard Disk

The hard disk is the most important component of a file server. The hard disk stores the files of the people who use the LAN. To a large extent, the reliability, access speed, and capacity of a server's hard disk determine whether people will be happy with the LAN and will use the LAN productively. The most common bottleneck in the average LAN is disk I/O time at the file server. And the most common complaint voiced by people on the average LAN is that the file server has run out of free disk space. Make sure that your file server's disk drives and hard disk controller are high-performance components, and that you have plenty of free disk space on your server's drives.

Evaluating the File Server CPU

The file server CPU tells the hard disk what to store and retrieve. The CPU is the next most important file server component after the hard disk. Unless your LAN will have only a few users and will never grow, a file server with a fast 80486 or Pentium CPU and plenty of RAM is a wise investment. The next section discusses server RAM.



The CPU chip in a computer executes the instructions given to it by the software you run. If you run an application, that application runs more quickly if the CPU is fast. Likewise, if you run a network operating system, that NOS runs more quickly if the CPU is fast.

Some network operating systems absolutely require certain types of CPU chips. NetWare Version 2, for example, requires at least an 80286 CPU. NetWare Versions 3 and 4 require at least an 80386. IBM LAN Server Version 2 and Microsoft LAN Manager Version 2 require that OS/2 1.3 be running on the server computer; OS/2 1.3 requires an 80286 or later CPU. LAN Server 3.0 requires that the file server use OS/2 2.x, which runs only on 80386 or later CPUs.

Evaluating Server RAM

The network operating system loads into the computer's RAM, just as any other application does. You need to have enough RAM in the computer for the NOS to load and run. On a peer LAN, a few megabytes of RAM might be enough, whereas on a server-based LAN, you might install 32M, 64M, or more in your file server.

You can realize significant performance gains with a faster CPU and extra RAM because of something called *caching*. If the file server has sufficient memory installed, it can "remember" those portions of the hard disk that it accessed previously. When the next user asks for the same file represented by those portions of the hard disk, the server can hand these to the next user without having to actually access the hard disk. Because the file server is able to avoid waiting for the hard disk to rotate into position, the server can do its job more quickly. The network operating system merely needs to look in the computer's RAM for the file data that a workstation has requested. Note that the network operating system's caching of file data is distinct from (and in addition to) any caching that might occur due to the hard disk or hard disk controller card having on-board memory.

Evaluating the Network Adapter Card

The server's network adapter card is the server's link to all the workstations on the LAN. All the requests for files enter the server through the network adapter, and all the response messages containing the requested files leave the server through the network adapter. Figure 11.9 shows a network adapter you might install in a file server. As you can imagine, a network adapter in a server is a busy component.

Fig. 11.9
The file server's network adapter sends and receives messages to and from all the workstations on the LAN.

All the network adapters on the LAN use Ethernet, Token Ring, ARCnet, or some other protocol. Within one of these protocols, however, you can find network adapters that perform better than others. A network adapter may be faster at processing messages because it has a large amount of on-board memory (RAM), because it contains its own microprocessor, or perhaps because the adapter uses one of the larger (longer) computer slots and thus can transfer more data between itself and the CPU at one time. A faster, more capable network adapter is an ideal candidate for installation in the file server.

Evaluating the Server's Power Supply

In a file server, the power supply is an important but often overlooked item. Power supply failures and malfunctions cause problems elsewhere in the computer and are difficult to diagnose. Your file server may display a message indicating that a RAM chip has failed, and then stop; the cause of the problem may indeed be a failed RAM chip, or the problem may be in the power supply.

The fan(s) in the power supply sometimes stop working or become obstructed with dust and dirt. The computer overheats and fails completely or acts strangely. Cleaning the fan(s)—after unplugging the computer from the wall outlet, of course—should be a part of the regular maintenance of your file server.

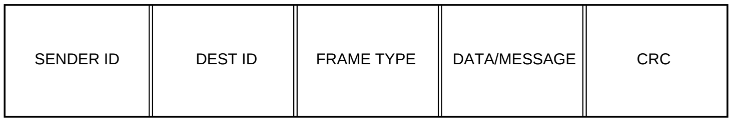
Power supplies vary considerably in quality. Some of the best power supplies are manufactured by the following company:

PC Power and Cooling
31510 Mountain Way
Bonsall, CA 92003
(800) 722-6555

Evaluating the Keyboard, Monitor, and Mouse

The keyboard, monitor, and mouse (if any) are not significant components on a file server computer. Often you can use lower quality, less expensive components for these parts. A typical file server runs unattended and may go for hours or days without interaction from you. You can power off the monitor for these long periods.

Note one caution about the keyboard: you should tuck the keyboard away so that falling objects (pencils or coffee mugs, for example) do not harm your network's file server.



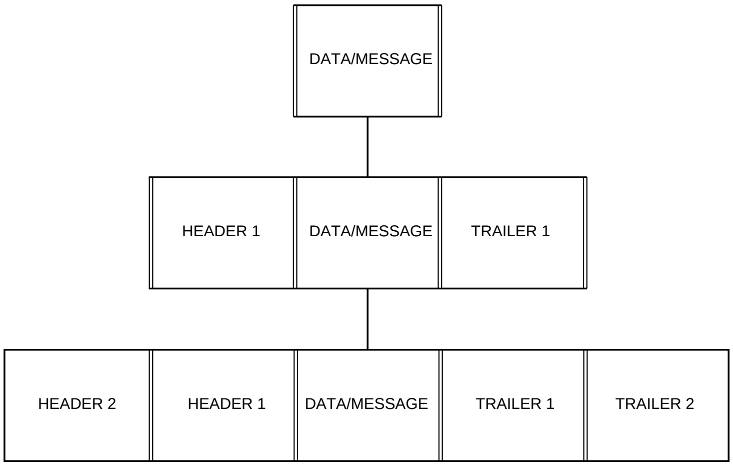
Examining Protocols, Frames, and Communications

The network adapter sends and receives messages among the LAN computers, and the cable carries the messages. The layer of protocols in each computer, however, turns the computers into a local area network.

At the lowest level, networked PCs communicate with one another and with the file server by using message packets, often called *frames*. These frames are the foundation on which all LAN activity is based. The network adapter, along with its support software, sends and receives these frames. Each computer has a unique address on the LAN to which frames can be sent.

You can send frames for various purposes, including the following:

- Opening a communications session with another adapter
- Sending data (perhaps a record from a file) to a PC
- Acknowledging the receipt of a data frame



- Broadcasting a message to all other adapters
- Closing a communications session

Figure 11.10 shows what a typical frame looks like. Different network implementations define frames in different ways, but the following data items are common to all implementations:

- The sender's unique network address
- The destination's unique network address
- An identification of the contents of the frame
- A data record or message
- A checksum or CRC for error-detection purposes

Fig. 11.10
The basic layout of a frame.

Using Frames That Contain Other Frames

The layering of protocols is a powerful concept. The lowest layer knows how to tell the network adapter to send a message, but that layer is ignorant of file servers and file redirection. The highest layer understands file servers and redirection but knows nothing about Ethernet or Token Ring. Together, though, the layers give you a local area network. Frames always are layered (see fig. 11.11).

When the higher level file redirection protocol gives a message to a midlevel protocol (such as NetBIOS, for example) and asks that the message be sent to another PC on the network (probably a file server), the midlevel protocol puts an *envelope* around the message packet and hands it to the lowest level protocol, implemented as the network support software and network adapter card. This lowest layer, in turn, wraps the NetBIOS envelope in an envelope of its own and sends it out across the network. In figure 5.11, you see each envelope labeled *header* and *trailer*. On receipt, the network support software on the receiving computer removes the outer envelope and hands the result upward to the next higher level protocol. The midlevel protocol running on the receiver's computer removes its envelope and gives the message—now an exact copy of the sender's message—to the receiving computer's highest-level protocol.

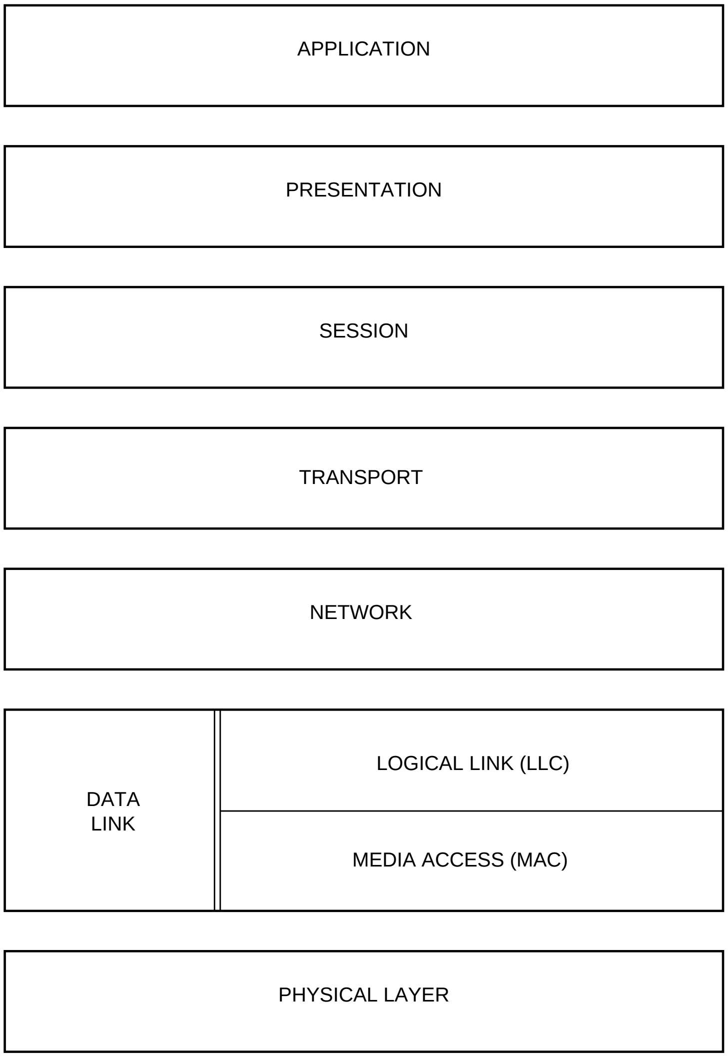


Fig. 11.11
Frame layers.

Different vendors split the LAN communications functions in different ways, but they all compare themselves to the OSI model.

Using the OSI Model

ISO, the International Standards Organization, has published a standard called the Open System Interconnection (OSI) model. Most vendors of LAN products endorse the OSI standard but have not yet implemented OSI fully. The OSI model divides LAN communications into seven layers. Most network operating system vendors use three or four layers of protocols.

The OSI model describes how communications between two computers should occur. Sometime in this decade, this theoretical standard will become a practical one as more and more vendors switch to OSI. The OSI model declares seven layers and specifies that each layer be insulated from the others by a well-defined interface. Figure 11.12 shows the seven layers.

Descriptions of the seven layers follow:

- *Physical*. This part of the OSI model specifies the physical and electrical characteristics of the connections that make up the network (twisted pair cables, fiber optic cables, coaxial cables, connectors, repeaters, and so on). You can think of this layer as the hardware layer. Although the rest of the layers may be implemented as chip-level functions rather than as actual software, the other layers are software in relation to this first layer.

- *Data Link*. At this stage of processing, the electrical impulses enter or leave the network cable. The network's electrical representation of your data (bit patterns, encoding methods, and tokens) is known to this layer, and only to this layer. It is at this point that errors are detected and corrected (by requesting retransmissions of corrupted packets). Because of its complexity, the Data Link layer often is subdivided into a Media Access Control (MAC) layer and a Logical Link Control (LLC) layer. The MAC layer deals with network access (token-passing or collision-sensing) and network control. The LLC layer, operating at a higher level than the MAC layer, is concerned with sending and receiving the user data messages.

Fig. 11.12
The OSI model.

- *Network*. This layer switches and routes the packets as necessary to get them to their destinations. This layer is responsible for addressing and delivering message packets.

- *Transport*. When more than one packet is in process at any time, the Transport layer controls the sequencing of the message components and regulates inbound traffic flow. If a duplicate packet arrives, this layer recognizes it as a duplicate and discards it.

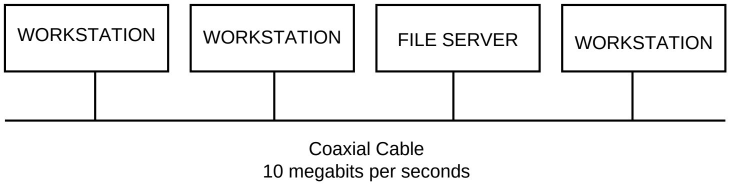
- *Session*. The functions in this layer enable applications running at two workstations to coordinate their communications into a single session (which you can think of in terms of a highly structured dialog). The Session layer supports the creation of the session, the management of the packets sent back and forth during the session, and the termination of the session.
- *Presentation*. When IBM, Apple, DEC, NeXT, and Burroughs computers want to talk to one another, obviously a certain amount of translation and byte reordering needs to be done. The Presentation layer converts data into (or from) a machine's native internal numeric format.
- *Application*. This is the layer of the OSI model seen by an application program. A message to be sent across the network enters the OSI model at this point, travels downward toward layer 1 (the Physical layer), zips across to the other workstation, and then travels back up the layers until the message reaches the application on the other computer through its own Application layer.

One of the factors that makes the network operating system of each vendor proprietary (as opposed to having an *open architecture*) is the vendor's noncompliance with the OSI model.

Using Low-Level Protocols

Local area networks work in one of two basic ways: collision-sensing or token-passing. Ethernet is an example of a collision-sensing network; Token Ring is an example of a token-passing network.

The Institute of Electrical and Electronic Engineers (IEEE) has defined and documented a set of standards for the physical characteristics of both collision-sensing and token-passing networks. These standards are known as IEEE 802.3 (Ethernet) and IEEE 802.5 (Token Ring). Be aware, though, that there are minor differences between the frame definitions for true Ethernet and for true IEEE 802.3. In terms of the standards, IBM's 16-mbps Token Ring adapter card is an 802.5 Token Ring extension. You learn the definitions and layout of Ethernet and Token Ring frames in the sections "Using Ethernet" and "Using Token Ring," later in this chapter.



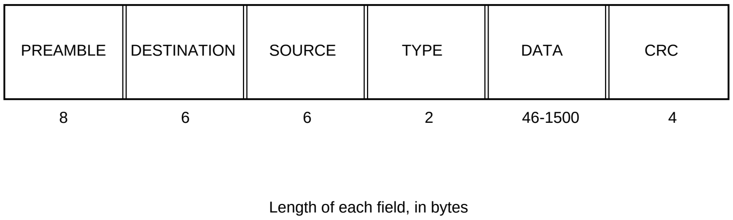
Some LANs don't conform to IEEE 802.3 or IEEE 802.5, of course. The most popular of these is ARCnet, available from such vendors as Datapoint Corporation, Standard Microsystems, and Thomas-Conrad. Other types of LANs include StarLan (from AT&T),

VistaLan (from Allen-Bradley), LANtastic (from Artisoft), Omninet (from Corvus), PC Net (from IBM), and ProNet (from Proteon).

Fiber Distributed Data Interface (FDDI) is a new physical-layer LAN standard. FDDI uses fiber optic cable and a token-passing scheme similar to IEEE 802.5 to transmit data frames at a snappy 100 mbps.

Using Ethernet

In the collision-sensing environment, often referred to by the abbreviation CSMA/CD (carrier sense, multiple access, with collision detection), the network adapter card listens to the network when it has a frame to send. If the adapter hears that another card is sending a frame at that moment, the card waits a moment and tries again. Even with this approach, collisions (two workstations attempting to transmit at exactly the same moment) can and do occur. It is the nature of CSMA/CD networks to expect collisions and to handle them by retransmitting frames as necessary. These retransmissions are handled by the adapter card and are not seen or managed by you or your applications. Collisions generally happen and are handled in less than a microsecond.



Caution

Although many people blame poor CSMA/CD (Ethernet) network performance on the number of users currently on the network who are sending and receiving message traffic, the truth is that more than 90 percent of transmission problems on an Ethernet network are the result of faulty cables or malfunctioning adapter cards.

On an Ethernet network, data is broadcast throughout the network in all directions at the rate of 10 mbps. All machines receive every frame, but only those meant to receive a frame (by virtue of the frame's destination network address) respond with an acknowledgment. Figure 11.13 illustrates an Ethernet network.

Fig. 11.13
An Ethernet network.

Ethernet is a LAN standard that is based on the Experimental Ethernet network designed and built in 1975 by Xerox at the Palo Alto Research Center (PARC). Ethernet operates at 10 mbps over 50-ohm coaxial cable; the current version is 2.0, established in November, 1982.

IEEE 802.3 is a LAN standard similar to Ethernet. The first edition of IEEE 802.3 was published in 1985. The differences between the two Ethernet standards are in the areas of network architecture and frame formats.

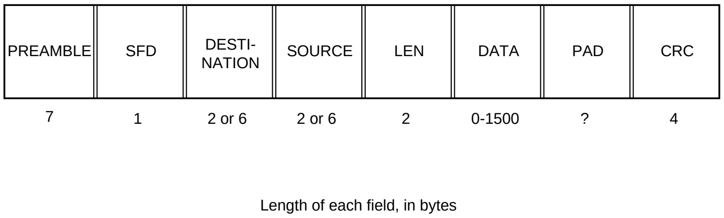
In terms of network architecture, IEEE 802.3 distinguishes between MAC and LLC layers; true Ethernet lumps these layers together into a single Data Link layer. Ethernet also defines an Ethernet Configuration Test Protocol (ECTP) that is absent from the IEEE 802.3 standard. Note, however, that the important differences between the two are in the types and lengths of the fields that make up a frame. These differences can cause the two protocols to be incompatible. The Ethernet and IEEE 802.3 frames are discussed in the following sections.

Using Ethernet Frames. Figure 11.14 shows the layout and data field definitions for a true Ethernet frame (the original, non-IEEE Ethernet).

Fig. 11.14
An Ethernet frame.

Descriptions of the original Ethernet frame follow:

- *Preamble*. This field, used for synchronization and framing, is 8 bytes long (the standard refers to a byte as an *octet*, or *8 bits*; you can call them *bytes*). The Preamble always contains the bit pattern 10101010 in the first 7 bytes, with 10101011 in the last (8th) byte.
- *Destination Address*. This field is 6 bytes in size and contains the address of the workstation that will receive this frame. The first (leftmost) bit of the first byte has a special meaning. If the leftmost bit is a 0, the destination address is a physical address that is unique throughout the Ethernet universe. As a result of a naming scheme administered by Xerox Corporation, the first three bytes are a group address assigned by Xerox, and the last three are assigned locally. If the leftmost bit is a 1, it represents a broadcast frame. In this case, the rest of the destination address can refer to a group of logically related workstations or to all workstations on the network (all 1s).



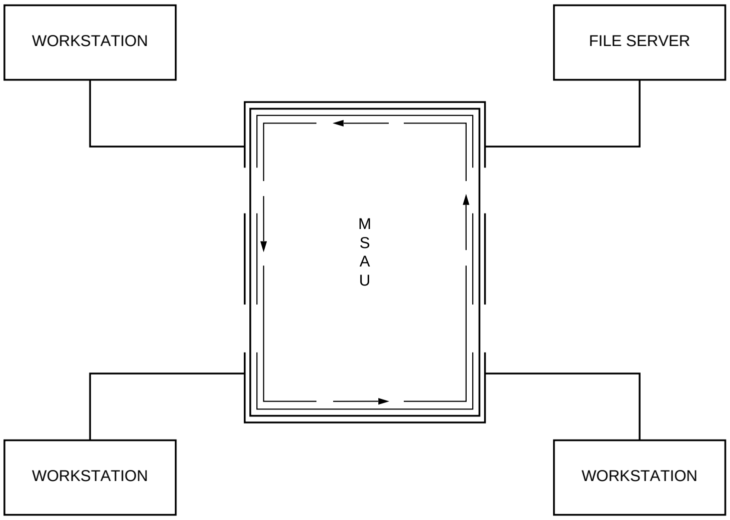
- *Source Address*. This address field is also 6 bytes, and it identifies the workstation sending the frame. The leftmost bit of the first byte always is 0.

- *Type*. This field contains 2 bytes of information that identify the type of the higher-level protocol that issued (or wants to receive) this frame. The Type field is assigned by Xerox and is not interpreted by Ethernet. It enables multiple high-level protocols (referred to as *Client layers*) to share the network without running into one another's messages.
- *Data Portion*. This portion of the frame can contain 46 to 1,500 bytes. It is the data message that the frame is intended to carry to the destination.
- *CRC*. Finally, the frame contains 4 bytes of cyclic redundancy checksum remainder, calculated via a CRC-32 polynomial. The workstation that receives this frame performs its own CRC-32 calculation on the frame and compares the calculated value to the CRC field in the frame to find out whether the frame arrived intact or was damaged in transit.

Disregarding the preamble for a moment, you can see that an entire Ethernet frame is between 64 and 1,518 bytes in size. You can see also that the minimum size of a data message is 46 bytes.

Using IEEE 802.3 Frames. Figure 11.15 shows an IEEE 802.3 frame, which contains the following fields:

- *Preamble*. This field contains 7 bytes of synchronization data. Each byte is the same bit pattern: 10101010.
- *Start Frame Delimiter (SFD)*. The SFD consists of a single byte that has the bit pattern 10101011. (The Preamble and SFD IEEE 802.3 fields match the single Ethernet Preamble field.)



- *Destination Address*. This field can contain 2 or 6 bytes, depending on which type of IEEE 802.3 network you install, and indicates the workstation for which the frame is intended. Note that all addresses on a particular network must be 2- or 6-byte addresses. The most popular type of IEEE 802.3, called 10BASE5, specifies 6-byte addresses. The first bit of the destination address is the individual/group bit. The I/G bit has a value of 0 if the address refers to a single workstation, or a 1 if it represents a group of workstations (a broadcast message). If the destination address is a 2-byte field, the rest of the bits form a 15-bit workstation address. If the destination address is a 6-byte field, however, the bit following the I/G bit is a universally/locally administered bit (the U/L bit). The U/L bit is a 0 for universally administered (global) addresses and is a 1 for locally administered addresses. The rest of the 6-byte field is a 46-bit workstation address.

Fig. 11.15
An IEEE 802.3 frame.

- *Source Address*. This is the 2- or 6-byte address of the sending workstation. The I/G (first) bit is always 0.
- *Length*. These two bytes express the length of the data portion of the frame.
- *Data Portion*. This field ranges from 0 to 1,500 bytes of data. If this field is less than 46 bytes, the next field (PAD) is used to fatten the frame to an acceptable (minimum) size.
- *Pad*. The Pad field contains enough bytes of filler to ensure that the frame has at least a certain overall size. If the data portion is large enough, the Pad field does not appear in the frame (Pad has zero length).
- *CRC*. The cyclic redundancy checksum remainder has 4 bytes of remainder from the CRC-32 algorithm—the same as for Ethernet.

The size of a frame under both true Ethernet and IEEE 802.3 Ethernet (assuming Type 10BASE5), excluding the Preamble and SFD, is the same: from 64 to 1,518 bytes. Under IEEE 802.3, however, it is permissible for the application (or an upper layer protocol) to send a data area that is less than 46 bytes, because the frame is padded automatically by the MAC layer. Under true Ethernet, data frames that are too small are considered to be error situations.

Using Token Ring

You can think of a token-passing network as a ring. Even though the network may be wired electrically as a star, data frames move around the network from workstation to workstation in ring fashion, as shown in figure 11.16. A workstation sends a frame to the MSAU (multistation access unit), which routes the frame to the next workstation.

Fig. 11.16
A token ring network.

Each network adapter card receives a frame from its upstream neighbor, regenerates the electrical signals making up the frame, and passes the result along to the next (downstream) workstation. The frame may consist of some data that one computer is sending to another, or the frame may be a token. A *token* is a 3-byte message, indicating that the LAN is idle.

When a workstation wants to send a frame, the network adapter waits for the token. The adapter then turns the token into a data frame containing a protocol-layered message.

The frame travels along from adapter to adapter until it reaches its destination, which acknowledges reception of the frame by setting certain bits in the frame. The data frame continues its journey around the ring. When the sending station receives its own frame back, and if the frame was properly received, the sender relinquishes use of the LAN by putting a new token into circulation. A token-passing network is designed so that collisions never occur.

Early Token Release

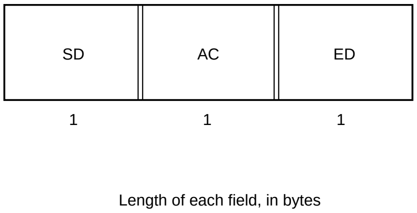
Early Token Release (ETR) easily is misunderstood; Token Ring itself is a fairly complex subject. On a momentarily idle Token Ring LAN, workstations circulate a token. The LAN becomes busy (carries information) when a workstation receives a token and turns it into a data frame targeted at the file server (or targeted back at a file-needy workstation if originated by a server that is answering a file I/O request). After receipt by its target node, the data frame continues circulating around the LAN until it reaches its source node. The source node turns the data frame back into a token that circulates until a downstream node needs it. So far, so good—these are just standard Token Ring concepts.

A workstation needs to send only a few bytes to tell the file server that it needs some part of a file. If the signal must go into and out of many workstations to circulate the ring, and if the data frame is small, latency occurs. *Latency* is the unproductive delay that occurs while the source node waits for its upstream neighbor to return its data frame.

The source node appends idle characters onto the LAN following the data frame until the data frame circulates the entire LAN and arrives back at the source node. The typical latency of a 4-mbps ring is about 50 to 100 idle characters. On a 16-mbps ring, latency may reach 400 or more bytes' worth of LAN time.

With Early Token Release, available only on 16-mbps networks, the originating workstation transmits a new token immediately after sending its data frame. Downstream nodes pass along the data frame and then receive an opportunity to transmit data themselves—the new token. If you had a Token Ring microscope, you would see tokens and other data frames (instead of a long trail of idle characters) chasing the data frame. You would also know that your 16-mbps Token Ring LAN is using ETR to keep itself busy.

Using Token Ring Frames. In 1985, Texas Instruments and IBM jointly developed the TMS380 chipset (although IBM doesn't use the chipset; it builds its own proprietary chipset, which is mostly compatible with the TI/IBM set). The TMS380 chipset



implements the IEEE 802.5 standards for the Physical and Data Link layers of the OSI model. The functions of both the MAC sublayer and the LLC sublayer of the Data Link layer are supported. Originally released as a set of five chips, the TI product now can be produced as a single chip.

Here are the TMS380 functions:

- *TMS38051 and 38052 chips*. These chips handle the lowest level: the ring interface itself. They perform the actual transmission/reception of data (frames), monitor cable integrity, and provide clocking functions.
- *TMS38020 chip*. This chip is the protocol handler. It controls and manages the 802.5 protocol functions.
- *ROM chip*. This chip has program code burned into it. The permanently stored software performs diagnostic and management functions.
- *TMS38010 chip*. This chip is a 16-bit dedicated microprocessor for handling communications; it executes the code in the ROM chip and has a 2.75K RAM buffer for temporary storage of transmitted and received data.

Although most people think of a Token Ring as a single piece of cable that all the workstations tap into, a Token Ring actually consists of individual point-to-point linkages. Joe’s workstation sends the token (or a data frame) to your workstation, your workstation sends the token or frame downstream to the next workstation, and so on. Only the fact that one of your downstream neighbors also happens to be Joe’s upstream neighbor makes it a ring. From a communications standpoint, the messages go directly from one PC to another.

Not all workstations on the ring are peers, although the differences are invisible to the outside world. One of the workstations is designated as the *active monitor*, which means that it assumes additional responsibilities for controlling the ring. The active monitor maintains timing control over the ring, issues new tokens (if necessary) to keep things going, and generates diagnostic frames under certain circumstances. The active monitor is chosen at the time the ring is initialized and can be any one of the workstations on the network. If the active monitor fails for some reason, there is a mechanism by which the other workstations (*standby monitors*) can decide which one becomes the new active monitor.

Three formats are defined for IEEE 802.5 Token Ring message packets: tokens, frames, and abort sequences. These formats are discussed in the following sections.

Using the Token. Figure 11.17 shows the first format of the IEEE 802.5 message packet: the token. In principle, the token is not a frame but simply a means by which each workstation can recognize when its turn to transmit has arrived.

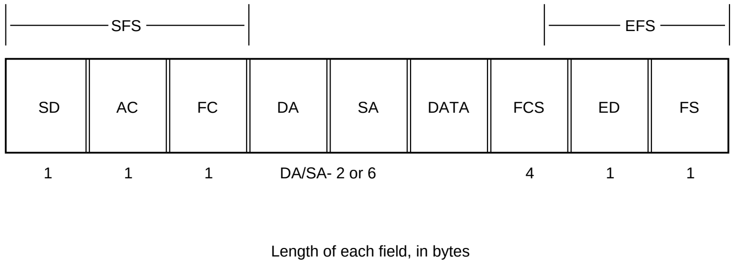
Fig. 11.17
A token.

A token is 3 bytes long (24 bits) and contains the following three fields:

- Start Delimiter
- Access Control
- End Delimiter

The Start Delimiter (SD) field appears at the beginning of the token (as well as at the beginning of every message or frame that is sent across the network). The SD field consists not of just 0s and 1s, but of a unique series of electrical impulses that cannot be mistaken for anything other than a Start Delimiter field. Because the SD field contains four nondata symbols (each 1 bit long) and four (normal) 0 bits in the field, the field totals 1 byte in size.

Next, comes the Access Control (AC) field. This field is divided into four subfields:



P P P T M R R R

P P P are the priority bits, T is the token bit, M is the monitor bit, and R R R are the reservation bits.

A network adapter can prioritize a token or frame by setting the priority bits to a value ranging from 0 to 7 (with 7 being the highest priority). A workstation can use the network (that is, change a token into a frame) only if it receives a token with a priority less than or equal to the workstation's own priority. The workstation's network adapter sets the priority bits to indicate the priority of the current frame or token. Refer to the description of the reservation bits for more on how this works.

The token bit has a value of 1 for a token and has a value of 0 for a frame.

The monitor bit is set to 1 by the active monitor and set to 0 by any workstation transmitting a token or frame. If the active monitor sees a token or frame that contains a monitor bit of 1, it knows that this token or frame has been once around the ring without being processed by a workstation. Because a sending workstation is responsible for removing its own transmitted frames (by recirculating a new token), and because high-priority workstations are responsible for grabbing a token that they claimed previously, the active monitor detects that something is wrong if a frame or a prioritized token has circulated the ring without having been processed. The active monitor cancels the transmission and circulates a new token.

The reservation bits work hand in hand with the priority bits. A workstation can place its priority in the reservation bits (if its priority is higher than the current value of the reservation bits). The workstation then has reserved the next use of the network. When a workstation transmits a new token, the workstation sets the priority bits to the value that it found in the RRR field of the frame it just received. Unless preempted by an even higher priority workstation, the workstation that originally set the reservation bits will be the next station to turn the token into a frame.

The final field of the token is the End Delimiter (ED) field. As with the Start Delimiter field, this field contains a unique combination of 1s and special nondata symbols that cannot be mistaken for anything else. The ED field appears at the end of each token. Besides marking the end of the token, the ED field contains two subfields: the Intermediate Frame bit and the Error-Detected bit. These fields are discussed in the next section; they pertain more to frames than to tokens.

Using the Data Frame. Figure 11.18 shows the second format of the IEEE 802.5 message packet: the true data frame. Data frames can, of course, contain messages that a network operating system or an application sends to another computer on the ring. Data frames also sometimes contain internal messages used privately among the Token Ring network adapter cards for ring-management purposes.

Fig. 11.18
A token ring data frame.

A frame consists of several groups of fields: the Start Frame Sequence (SFS), the Destination Address (DA), the Source Address (SA), the data itself (DATA), the Frame Check Sequence (FCS), and the End Frame Sequence (EFS). Together, these fields form a message record (envelope) that is used to carry ring-management information (MAC data) or user data (LLC data). You already know about LLC data; these are the frames that contain application-oriented data, such as PC-to-PC messages or a portion of a disk file (from a file server) that is being shared through the network operating system. The network adapters use MAC frames internally, though, to control and manage the ring. The IEEE 802.5 standard defines six MAC control frames. The Frame Control field indicates the type of the frame (MAC or LLC); if it is a MAC frame, this field also indicates which of the six frames is represented by this particular frame.

The six MAC frames follow:

- *Duplicate Address Test*. Sent by a workstation when it joins the ring to ensure that its address is unique.
- *Active Monitor Present*. Circulated every so often by the active monitor to let other workstations know that it is still alive.
- *Standby Monitor Present*. Sent by a monitor other than the active monitor.
- *Claim Token*. If a standby monitor thinks that the active monitor may have died, it starts sending Claim Token frames. The standby monitors then go through a process of negotiation with one another to determine which one becomes the new active monitor.
- *Beacon*. Sent in the event of a major network problem, such as a broken cable or a workstation that is transmitting without waiting for the token. By detecting which station is sending the Beacon frame, diagnostic software can localize the problem.
- *Purge*. Sent after ring initialization and after a new active monitor establishes itself.

Each frame (MAC or LLC) begins with a start frame sequence, which contains three fields:

- *Start Delimiter (SD)*. The definition of SD is the same for frames as for tokens.
- *Access Control (AC)*. The definition is the same for frames as for tokens.
- *Frame Control (FC)*. This is a 1-byte field containing two subfields—Frame Type and MAC Control ID:

  F F C C C C C C

The two frame type bits (FF) have a value of 00 for MAC frames and 01 for LLC frames (11 and 10 are reserved).

The MAC control ID bits identify the type of ring-management frame:

CCCCCC	MAC Frame
000011	Claim Token
000000	Duplicate Address Test
000101	Active Monitor Present
000110	Standby Monitor Present
000010	Beacon
000100	Purge

The Destination Address (DA) follows the Start Frame Sequence fields. The DA field can be 2 or 6 bytes. With 2-byte addresses, the first bit indicates whether the address is a group address or an individual address (just as in the collision-sensing IEEE 802.3 protocol). With 6-byte addresses, the first bit also is an I/G bit, and the second bit tells whether the address is assigned locally or globally (the U/L bit, which again is the same as in the IEEE 802.3 protocol). The rest of the bits form the address of the workstation to which the frame is addressed.

The Source Address (SA) field is the same size and format as the Destination Address field.

The data portion of the frame (DATA) can contain a user data message record intended for (or received from) a midlevel protocol such as IPX, TCP/IP, or NetBIOS. Or the Data field can contain one of the MAC frames just discussed. The Data field has no specified maximum length, although there are practical limits on its size based on how long a single workstation may have control of the ring.

The Frame Check Sequence (FCS) field is 4 bytes of remainder from the CRC-32 cyclic redundancy checksum algorithm. It is used for error detection.

Length of each field, in bytes

The End Frame Sequence (EFS) is composed of two fields: the End Delimiter and the Frame Status. Descriptions of these fields follow:

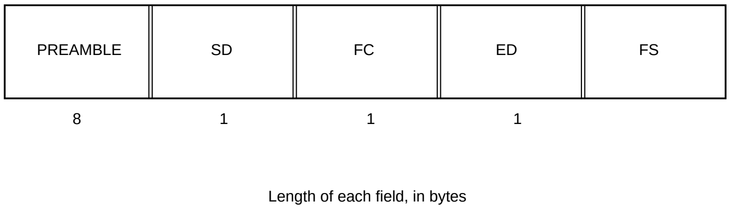
- *End Delimiter (ED)*. You read about this field in relation to tokens; in a frame, however, this field takes on additional meaning. Besides consisting of a unique pattern of electrical impulses, it contains two subfields—each 1 bit in size. The intermediate frame bit is set to 1 if this frame is part of a multiple-frame transmission, and the bit is set to 0 if the frame is the last (or only) frame. The error-detected bit starts as a 0 when a frame is sent originally. Each workstation's network adapter, as it passes the frame along, checks for errors (verifying that the CRC in the Frame Check Sequence field still corresponds to the contents of the frame, for example). An adapter sets the error-detected bit to 1 if the adapter finds something wrong. The intervening network adapters that see an already set error-detected bit pass the frame along. The originating adapter notices that a problem occurred and tries again by retransmitting the frame.
- *Frame Status (FS)*. This 1-byte field contains four reserved bits (R) and two subfields—the address-recognized bit (A) and the frame-copied bit (C):

  A C RR A C RR

Because the calculated CRC does not encompass the Frame Status field, each of the 1-bit subfields is duplicated within frame status to ensure data integrity. A transmitting workstation sets the address-recognized bit to 0 when it originates a frame; the receiving workstation sets this bit to 1 to signal that it has recognized its destination address. The frame-copied bit also starts out as 0 but is set to 1 by the receiving (destination) workstation when it copies the contents of the frame into its own memory (when it actually receives the data). The data is copied (and the bit set) only if the frame is received without error. If the originating (source) workstation gets its frame back with both of these bits set, it knows that a successful reception occurred.

If, however, the address-recognized bit is not set by the time the frame gets back to the originating workstation, the destination workstation is no longer on the network; the other workstation must have crashed or powered off suddenly.

Another situation occurs when the destination address is recognized but the frame-copied bit is not set. This setting tells the originating workstation that the frame got damaged in transit (the error-detected bit in the end delimiter also will be set).



If the address-recognized bit and the frame-copied bit are both set, but the error-detected bit also is set, the originating workstation knows that the error occurred after the frame was received correctly.

Using the Abort Sequence. Figure 11.19 shows the third format of the IEEE 802.5 message packet: the abort sequence. An abort sequence can occur anywhere in the bit stream and is used to interrupt or terminate the current transmission.

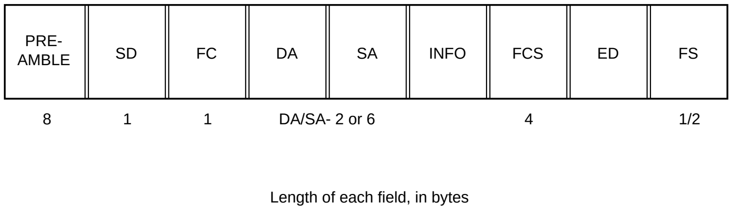


Fig. 11.19
An abort sequence.

An abort sequence consists of a Start Delimiter followed by an End Delimiter. An abort sequence signals cancellation of the current frame or token transmission.

Using the Fiber Distributed Data Interface (FDDI)

The Fiber Distributed Data Interface (FDDI) is a much newer protocol than Ethernet or Token Ring. Designed by the X3T9.5 Task Group of ANSI (the American National Standards Institute), FDDI passes tokens and data frames around a ring of optical fiber at a rate of 100 mbps. FDDI was designed to be as much like the IEEE 802.5 Token Ring standard as possible. Differences occur only where necessary to support the faster speeds and longer transmission distances of FDDI.

If FDDI were to use the same bit-encoding scheme used by Token Ring, every bit would require two optical signals: a pulse of light and then a pause of darkness. This means that FDDI would need to send 200 million signals per second to have a 100-mbps transmission rate. Instead, the scheme used by FDDI—called 4B/5B—encodes 4 bits of data into 5 bits for transmission so that fewer signals are needed to send a byte of information. The 5-bit codes (symbols) were chosen carefully to ensure that network timing requirements are met. The 4B/5B scheme, at a 100-mbps transmission rate, actually causes 125 million signals per second to occur (this is 125 megabaud). Also, because each carefully selected light pattern symbol represents 4 bits (a half byte, or *nibble*), FDDI hardware can operate at the nibble and byte level rather than at the bit level, making it easier to achieve the high data rate.

Two major differences in the way the token is managed by FDDI and IEEE 802.5 Token Ring exist. In Token Ring, a new token is circulated only after a sending workstation gets back the frame that it sent. In FDDI, a new token is circulated immediately by the sending workstation after it finishes transmitting a frame. FDDI doesn't use the Priority and Reservation subfields that Token Ring uses to allocate system resources. Instead, FDDI classifies attached workstations as *asynchronous* (workstations that are not rigid about the time periods that occur between network accesses) and *synchronous* (workstations having very stringent requirements regarding the timing between transmissions). FDDI uses a complex algorithm to allocate network access to the two classes of devices.

Figure 11.20 shows an FDDI token. The token consists of Preamble, Start Delimiter, Frame Control, End Delimiter, and Frame Status fields. These fields have the same definition for tokens as for frames.

Fig. 11.20
An FDDI token.

Figure 11.21 shows the layout of an FDDI frame. Notice the similarity to the IEEE 802.5 Token Ring frames just discussed. An FDDI frame, like its slower cousin, carries MAC control data or user data.

Fig. 11.21
An FDDI frame.

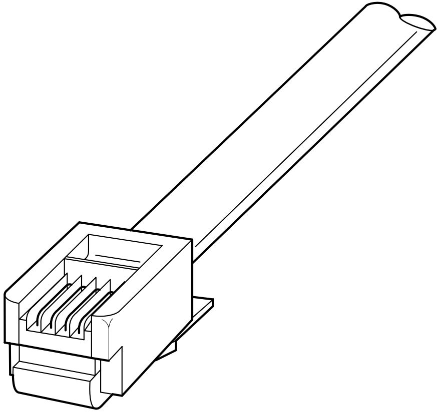
The fields in an FDDI frame follow:

- *Preamble*. This field is used for synchronization purposes. Although this field is initially 64 bits (16 symbol-encoded nibbles) in size, subsequent workstations can modify the Preamble's length dynamically according to their own clocking and synchronization requirements.
- *Start Delimiter (SD)*. A unique two-symbol (1-byte) field; its pattern identifies the start of the frame.

- *Frame Control (FC)*. A two-symbol (1-byte) field made up of the following subfields:

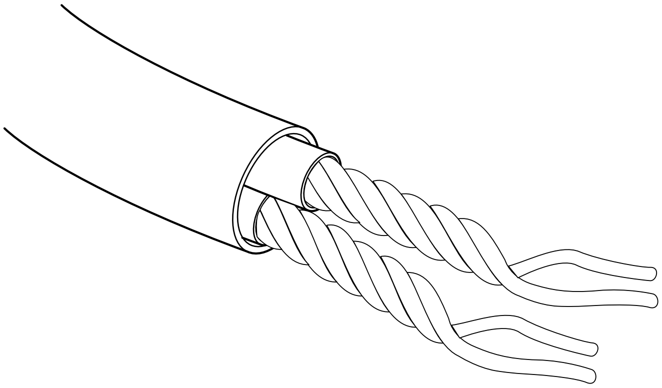
  C L FF TTTT

  The C subfield designates the frame class, which tells whether the frame is being used for synchronous or asynchronous service. The L bit is the frame address length and indicates whether 16- or 48-bit addresses are being used (unlike with Ethernet and Token Ring, both kinds of addresses are possible on the same FDDI network). The FF bits are the Frame Format subfield, and express whether the frame is a MAC frame carrying ring-management information or an LLC frame carrying user data. If it is a MAC frame, the T T T T bits specify the type of the MAC control frame contained in the Info field.



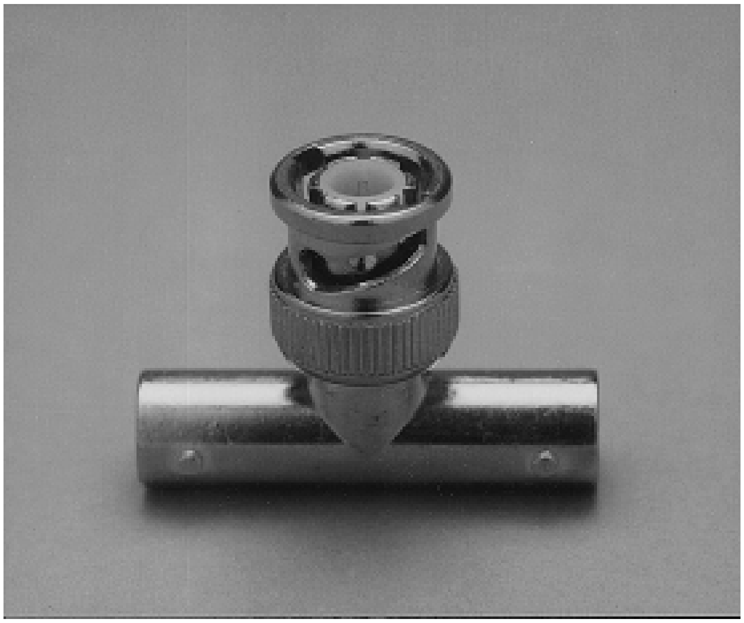
Unshielded twisted pair

- *Destination Address (DA)*. This field can be 16 bits or 48 bits and identifies the workstation to which this frame is being sent.

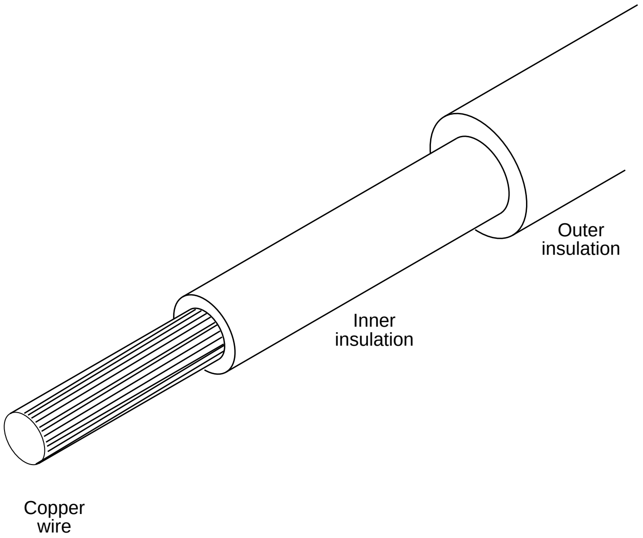


- *Source Address (SA)*. This field, which can be 16 or 48 bits, identifies the sending workstation.

- *Information (INFO)*. This field is the data portion of the frame. It contains a MAC control record or user data. This field can vary in length, but it cannot cause the overall length of the frame to exceed 4,500 bytes.
- *Frame Check Sequence (FCS)*. This field contains 4 bytes (8 symbols) of CRC data used for error-checking.
- *End Delimiter (ED)*. In a frame, this field is 1 nibble (1 symbol) long. In a token, it is 1 byte (2 symbols) long. This field uniquely identifies the end of the frame or token.
- *Frame Status (FS)*. This field is of arbitrary length, and contains the error-detected bit, the address-recognized bit, and the frame-copied bit. These subfields do the same job on an FDDI network as on a Token Ring network.



Using LAN Cables



Cabling systems for LANs vary widely in their appearance, characteristics, intended purpose, and cost. This chapter discusses the three most popular ways to tie computers together on a LAN: the IBM cabling system, the AT&T premises distribution system, and the Digital Equipment Corporation cabling concept called DECconnect.

Generally speaking, the cabling systems described in the next few sections use one of three distinct cable types. These are twisted pair (shielded and unshielded) cable, coaxial cable (thin and thick), and fiber optic cable.

Using Twisted Pair Cable

Twisted pair cable is just what its name implies: insulated wires with a minimum number of twists per foot. Twisting the wires reduces electrical interference (*attenuation*). *Shielded twisted pair* refers to the amount of insulation around the wire and therefore, its noise immunity. You are familiar with unshielded twisted pair; it is often used by the telephone company. Shielded twisted pair, however, is entirely different in appearance. Shielded twisted pair looks somewhat like the wire used to carry house current (110 volts) throughout your home or apartment. Appearances are deceiving, because shielded twisted pair actually carries a relatively low voltage signal. The heavy insulation is for noise reduction, not safety.

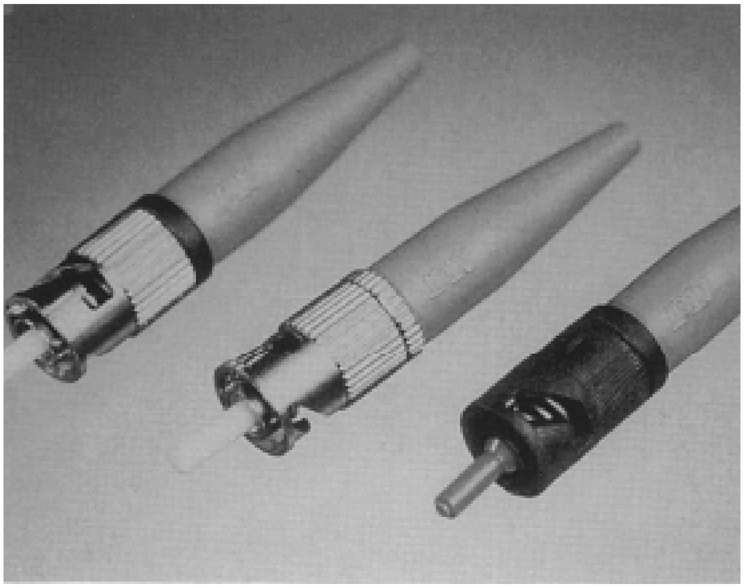


Figure 11.22 shows unshielded twisted pair cable; figure 11.23 illustrates shielded twisted pair cable.

Fig. 11.22
An unshielded twisted pair cable.

Fig. 11.23
A shielded twisted pair cable.

Using Coaxial Cable

Coaxial cable is fairly prevalent in your everyday life; you often find it connected to the backs of television sets and audio equipment. *Thin* and *thick*, of course, refer to the diameter of the coaxial cable. Standard Ethernet cable (thick Ethernet) is as thick as your thumb. The newer ThinNet (sometimes called CheaperNet) cable is about the size of your little finger. The thick cable has a greater degree of noise immunity, is more difficult to damage, and requires a *vampire tap* (a piercing connector) and a drop cable to connect to a LAN. Although thin cable carries the signal over shorter distances than the thick cable, ThinNet uses a simple BNC connector (a bayonet-locking connector for thin coaxial cables), is lower in cost, and has become a standard in office coaxial cable.

Figure 11.24 shows an Ethernet BNC coaxial connector, and figure 11.25 illustrates the design of coaxial cable.

Fig. 11.24
An Ethernet coaxial cable connector.

Fig. 11.25

Coaxial cable.

Using Fiber Optic Cable

Fiber optic cable, as its name suggests, uses light rather than electricity to carry information. Fiber can send data over huge distances at high speeds, but it is expensive and difficult to work with. Splicing the cable, installing connectors, and using the few available diagnostic tools for finding cable faults are skills that very few people have.

Fiber optic cable is simply designed, but unforgiving of bad connections. Fiber cable usually consists of a core of glass thread, with a diameter measured in microns, surrounded by a solid glass cladding. This, in turn, is covered by a protective sheath. The first fiber optic cables were made of glass, but plastic fibers also have been developed. The light source for fiber optic cable is a light-emitting diode (LED); information usually is encoded by varying the intensity of the light. A detector at the other end of the cable converts the received signal back into electrical impulses. Two types of fiber cable exist: single mode and multimode. Single mode has a smaller diameter, is more expensive, and can carry signals for a greater distance.

Figure 11.26 illustrates fiber optic cables and their connectors.

Fig. 11.26

Fiber optic cables use light to carry LAN messages. The ST connector is commonly used with fiber optic cables.

Using the IBM Cabling System

The IBM cabling system, ironically, is not manufactured or sold by IBM. This cabling system consists of a published IBM standard for wiring systems in office buildings that defines cabling system components and different cable types. When it was introduced in 1984, IBM described the IBM cabling system as the intended backbone of its Token Ring network. The first such cables to be manufactured by third-party companies were tested by IBM, verified to IBM specifications, and actually given IBM part numbers. At present, however, cable manufacturers have to rely on the ETL or UL independent testing laboratories or on industry-standard manufacturers (such as AMP) to verify compliance with the specifications published by IBM.

The IBM specification defines workstation faceplates, adapters/connectors, access units, and wiring-closet termination methods. The standard also defines the following cable types:

- *Type 1 data cable*. Copper-based, for data connections only. Available in nonplenum, plenum, and outdoor varieties. It consists of two twisted pairs of 22-gauge solid conductors, shielded with both foil and braid, and covered with a polyvinyl-chloride (PVC) sheath. Type 1 data cable is used for connecting terminal devices located in work areas to distribution panels located in wiring closets and for connecting between wiring closets. The plenum cable is installed in plenums, ducts, and spaces used for environmental air; in case of fire, it gives off less toxic fumes than nonplenum cable. The outdoor cable is protected in a corrugated metallic shield with a polyethylene sheath, and the core is filled with a jellylike compound to prevent moisture from entering.

- *Type 2 data and telephone cable*. For both data and voice (telephone) applications. This cable is similar to Type 1 but has four additional twisted pairs (22-gauge). Type 2 cable comes in plenum and nonplenum varieties.
- *Type 3 telephone twisted pair cable*. Consists of four-pair, 24-gauge wire in polyvinylchloride plastic. This cable is equivalent to the IBM Rolm specification and is available in plenum. This cable is unshielded and not as immune to noise as Type 1 cable when used for data.
- *Type 5 fiber optic cable*. Contains two 100/140-micron multimode optical fibers (100-micron core surrounded by 140-micron cladding layer). This cable is not defined by IBM.
- *Type 6 patch panel cable*. For connecting a workstation to a wall faceplate or making connections within a wiring closet. This cable is more flexible than Type 1 cable (hence, its use as patch cable). This cable consists of two twisted pairs of 26-gauge stranded conductors.
- *Type 8 undercarpet cable*. An undercarpet cable useful for open office or workstation areas where there are no permanent walls. Type 8 cable consists of two pairs of 26-gauge solid conductors in a flat sheath.
- *Type 9 low-cost plenum cable*. An economy version of Type 1 plenum cable, with a maximum transmission distance about two-thirds that of Type 1 cable. Type 9 cable consists of two twisted pairs of 26-gauge stranded conductors. This cable is not defined by IBM.

Connecting the Cables

In a token-passing network, the cables from the workstations (or from the wall faceplates) connect centrally to a multistation access unit (abbreviated MSAU, or sometimes just MAU). The MSAU keeps track of which workstations on the LAN are neighbors and which neighbor is upstream or downstream. It is an easy job; the MSAU usually does not even need to be plugged into a electrical power outlet. The exceptions to this need for external power are MSAUs that support longer cable distances, or the use of unshielded twisted pair (Type 3) cable in high-speed LANs. The externally powered MSAU helps the signal along by regenerating it.

An IBM MSAU has eight ports for connecting one to eight Token Ring devices. Each connection is made with a genderless data connector (as specified in the IBM cabling system). The MSAU has two additional ports, labeled RI (Ring-In) and RO (Ring-Out), that daisy-chain several MSAUs together when you have more than eight workstations on the LAN.

It takes several seconds to open the adapter connection on a Token Ring LAN (something you may have noticed). During this time, the MSAU and your Token Ring adapter card perform a small diagnostic check, after which the MSAU establishes you as a new neighbor on the ring. After being established as an active workstation, your computer is linked on both sides to your upstream and downstream neighbors (as defined by your position on the MSAU). In its turn, your Token Ring adapter card accepts the token or frame, regenerates its electrical signals, and gives the token or frame a swift kick to send it through the MSAU in the direction of your downstream neighbor.

In an Ethernet network, the number of connections (taps) and their intervening distances are limiting factors. Repeaters regenerate the signal every 500 meters or so. If repeaters were not used, *standing waves* (additive signal reflections) would distort the signal and cause errors. Because collision detection depends partly on timing, only five 500-meter segments and four repeaters can be placed in series before the propagation delay becomes longer than the maximum allowed period for the detection of a collision. Otherwise, the workstations farthest from the sender would be unable to determine whether a collision had occurred.

The people who design computer systems love to find ways to circumvent limitations. Manufacturers of Ethernet products have made it possible to create Ethernet networks in star, branch, and tree designs that overcome the basic limitations mentioned. You can have thousands of workstations on a complex Ethernet network.

Local area networks are local because the network adapters and other hardware components cannot send LAN messages more than about a few hundred feet. Table 11.9 reveals the distance limitations of different kinds of LAN cable. In addition to the limitations shown in the table, keep in mind that you cannot connect more than 30 computers on a ThinNet Ethernet segment, more than 100 computers on a ThickNet Ethernet segment, more than 72 computers on unshielded twisted pair Token Ring cable, or more than 260 computers on shielded twisted pair Token Ring cable.

Table 11.9 Network Distance Limitations

Network Adapter	Cable Type	Maximum	Minimum
Ethernet	Thin	607 ft.	20 in.
	Thick (drop cable)	164 ft.	8 ft.
	Thick (backbone)	1,640 ft.	8 ft.
	UTP	328 ft.	8 ft.
Token Ring	STP	328 ft.	8 ft.
	UTP	148 ft.	8 ft.
ARCnet (passive hub)		393 ft.	Depends on cable
ARCnet (active hub)		1,988 ft.	Depends on cable

Evaluating Fast Network Adapters

As mentioned earlier in this chapter, network adapters generally are collision-sensing or token-passing. A network adapter's design ties it to one of the low-level protocols—Ethernet, Token Ring, FDDI, ARCnet, or some other protocol.

If you have fast workstations and a fast file server, you want a fast network. Even 16 mbps may be too slow if your applications are data-intensive. TCNS, from Thomas-Conrad, operates at 100 mbps and doesn't cost much more than Token Ring. TCNS gives you all the advantages of FDDI without FDDI's high price tag. NetWare, LAN Manager, POWERLan, LANtastic, and other ARCnet-compatible network operating systems work well with TCNS. The only catch is that you have to use fast computers to realize performance gains with TCNS.

You can use the same shielded twisted pair (IBM Type 1) or coaxial (RG62A/U) cabling already in place for Token Ring or ARCnet, or you can install 62.5-micron fiber optic cable. You can mix and match cable types by using a TCNS Smart Hub with different connectors. You wire a TCNS network in a distributed star fashion, just as you would with ARCnet or Token Ring. TCNS adapters and hubs use ST connectors for fiber optic cable, BNC connectors for coaxial, and DB-9 connectors for STP.

A TCNS network adapter is register-compatible with an ARCnet adapter, which enables TCNSs to use industry-standard ARCnet software drivers. Thomas-Conrad also supplies "Accelerated Drivers" for an even greater performance boost. TCNS consists of network adapters with STP, coaxial, or fiber optic connectors; one or more Thomas-Conrad Smart Hubs; and software drivers. The adapters come in 16- and 32-bit, and ISA- and EISA-bus varieties. You can put up to 255 TCNS workstations on a single LAN segment, and you can span significant distances: 2,950 feet (hub to workstation) with fiber optic cable, 492 feet with shielded twisted pair cable, and 338 feet with RG62A/U coaxial cable.

Collision-sensing and token-passing adapters contain sufficient on-board logic to know when it is permissible to send a frame and to recognize frames intended for the adapters. With the adapter support software, both types of cards perform seven major steps during the process of sending or receiving a frame. Outbound, when data is being sent, the steps are performed in the order presented in the following list. Inbound, as data is received, however, the steps are reversed. Here are the steps:

1. *Data transfer*. Data is transferred from PC memory (RAM) to the adapter card or from the adapter card to PC memory via DMA, shared memory, or programmed I/O.

2. *Buffering*. While being processed by the network adapter card, data is held in a buffer. The buffer gives the card access to an entire frame at once, and the buffer enables the card to manage the difference between the data rate of the network and the rate at which the PC can process data.

3. *Frame formation*. The network adapter has to break up the data into manageable chunks (or, on reception, reassemble it). On an Ethernet network, these chunks

are about 1,500 bytes. Token Ring networks generally use a frame size of about 4K. The adapter prefixes the data packet with a frame header and appends a frame trailer to it. The header and trailer are the Physical layer's envelope, which you learned about earlier in this chapter. At this point, a complete, ready-for-transmission frame exists. (Inbound, on reception, the adapter removes the header and trailer at this stage.)

4. *Cable access*. In a CSMA/CD network such as Ethernet, the network adapter ensures that the line is quiet before sending its data (or retransmits its data if a collision occurs). In a token-passing network, the adapter waits until it gets a token it can claim. (These steps are not significant to receiving a message, of course.)
5. *Parallel/serial conversion*. The bytes of data in the buffer are sent or received through the cables in serial fashion, with one bit following the next. The adapter card does this conversion in the split second before transmission (or after reception).
6. *Encoding/decoding*. The electrical signals that represent the data being sent or received are formed. Most network adapters use *Manchester encoding*. This technique has the advantage of incorporating timing information into the data through the use of *bit periods*. Instead of representing a 0 as the absence of electricity and a 1 as its presence, the 0s and 1s are represented by changes in polarity as they occur in relation to very small time periods.
7. *Sending/receiving impulses*. The electrically encoded impulses making up the data (frame) are amplified and sent through the wire. (On reception, the impulses are handed up to the decoding step.)

Of course, the execution of all of these steps takes only a fraction of a second. While you were reading about these steps, thousands of frames could have been sent across the LAN.

Network Adapter cards and the support software recognize and handle errors, which occur when electrical interference, collisions (in CSMA/CD networks), or malfunctioning equipment cause some portion of a frame to be corrupted. Errors generally are detected through the use of a cyclic redundancy checksum (CRC) data item in the frame. The CRC is checked by the receiver; if its own calculated CRC doesn't match the value of the CRC in the frame, the receiver tells the sender about the error and requests retransmission of the frame in error. Several products exist that perform network diagnostic and analysis functions on the different types of LANs, should you find yourself in need of such troubleshooting.

The different types of network adapters vary not only in access method and protocol, but also in the following elements:

- Transmission speed
- Amount of on-board memory for buffering frames and data
- Bus design (8-bit, 16-bit, or MicroChannel)
- Bus speed (some fail when run at high speeds)
- Compatibility with various CPU chipsets