A Primer on Asynchronous Modem Communication

by
Winfield Stanton and Thomas Spencer

Last Updated: May 1998

This paper covers the basic RS-232 signaling, error correction, compression, and modulation standards.

In the near future, the asynchronous modem will continue to be an important means of transferring information to and from the personal computer. By some estimates, more than 60 percent of people who access the Internet do so with a conventional dial-up modem and a connection to an analog telephone line (www.boardwatch.com). You may be connecting to a bulletin board system (BBS) through a piece of communications software or you may be using the Windows NT® Remote Access Service (RAS) to dial out to your Internet Service Provider (ISP). Whether you use a simple file transfer protocol such as XMODEM or another protocol such as TCP/IP over the RAS Point-to-Point (PPP) protocol, the underlying asynchronous communications issues are the same.

Most asynchronous modem communication implemented today conforms to the RS-232 signaling standard (RS stands for recommended standard), and the most important standard layered onto the RS-232 signaling is the modulation standard. Other standards such as error correction and data compression also come into play.

Asynchronous communication is a method of serial data transfer that is common to a number of protocols, such as XMODEM, YMODEM, and Kermit. For the purpose of this paper, the set of rules common to asynchronous communication will occasionally be referred to as the *asynchronous protocol*. One of the main things to learn about the asynchronous protocol is how the modem and the computer use the different leads in the serial cable to communicate according to the RS-232 standard.

BASIC TERMINOLOGY

It is necessary to first clarify some of the basic terms important to this topic. For example, what are DTE and DCE? What are the differences between the *baud rate* and the speed of data transfer measured in bits per second (bps)? What does a modem actually do, and where did it get its name? And finally, what exactly is meant by a modulation standard, and what are the key elements that make up such a standard? These are the questions that will be addressed in the following sections.

Most discussions of networking protocols are framed against the protocol layers defined by the Open Systems Interconnect (OSI) model. Although a detailed discussion of that model is beyond the scope of this paper, it is important to know that the asynchronous protocol consists of just the application, data link, and physical layers of the OSI model. Also, the functions of the data link and physical layer overlap, because they are performing functions that are supposed to be performed by the other layer according to the OSI model. For example, the data-link layer works by changing electrical voltage levels of circuits (wires and pins) in the physical layer, which is a function of the physical layer in the OSI model.

Data terminal equipment (DTE) is the combination of a computer, a serial port, and application software that communicates with a second application—the *remote application*—over a telephone line. The remote application is also a configuration of a computer, a serial port, and an application that makes up the second DTE.

Most computers use a chip called a universal asynchronous receiver/transmitter (UART) to convert the computer's synchronous parallel data into asynchronous serial data ready for transmission to the modem.

The data communications equipment (DCE) is the modem. A DCE is also known as data circuit-terminating equipment, and its purpose is to link the DTE to the communication line.

Various asynchronous standards define the interface or signaling that goes on between the DTE and the DCE, and between the two DCEs. The connection between two modems is an ordinary telephone line. Because the connection is a *dial-up* connection, it is also called a *switched* line. For this reason, the telephone lines are known as a public switched telephone network (PSTN). MCI, Sprint, and AT&T are all PSTNs.

Sometimes the telephone lines are simply referred to as the plain old telephone system (POTS). An ordinary phone line has a physical limitation in transmitting data. The bandwidth of a an ordinary phone line lies within the range of approximately 300 to 3,000 Hz—a range that covers the most frequently used spoken tones, but that is far less than the full range of human hearing. PSTN bandwidth is comparable to less than half the playable notes on a standard piano.

This bandwidth limitation for telephone lines means that the maximum of 2,400 distinct tones per second can be transmitted on the line. This data transfer rate is measured as the baud rate. A baud is a single change of state in the analog signal per second. This change may be in frequency, amplitude, or phase of the carrier signal. Therefore, 2400 baud is the maximum throughput a PSTN will allow without the aid of some modulation scheme to increase the bps data throughput for the same maximum baud rate.

Sometimes a simple telephone line is a very fast physical link that is split at each end into many slower lines, each of which operates as an individual phone line. For example, a *statistical multiplexer* is one piece of equipment that splits a high-speed (T1) trunk line into many slower (9600 bps) lines so that each of the slower lines appears to have a dedicated modem attached to it.

Another variation on this same principle is the implementation of a *hunt group*. Using a hunt group over a high bandwidth line, one phone number is routed to a series of 5, 10, or more different inbound extensions, each of which has an individual modem. The hunt group finds the first available modem from the bank of modems and connects the caller to that modem. This process continues for each successive caller until all available modems are in use. All subsequent callers get a busy signal until one of the modems is free to take a call.

Modems take digital input from terminals or computers and map the discreet bits onto an analog sine wave (the carrier wave). The carrier and the modulated signal are transmitted to the remote modem, which strips off the signal riding the carrier, converts it back into a digital bit stream, and then sends the re-digitized data stream to the remote computer or terminal. The process is called *modulation-demodulation*, from which the modem takes its name.

There are five or six major schemes called modulation standards for mapping or unmapping the digital signal onto or from the carrier frequency. A discussion of these modulation standards is beyond the scope of this paper; however, the modulation standard attributes discussed in the next section are important to understand.

Modulation Standard Attributes

The frequency of the carrier sine wave, how many carrier waves there actually are, the physical modulation technique, and the speed at which the two modems can modulate or demodulate the digital signal are all attributes of the modulation standard. For two modems to communicate, they must be able to first negotiate to a common modulation standard.

Notice that the modulation technique is not equivalent to the modulation standard. The following is a list of some of the most commonly used modulation techniques:
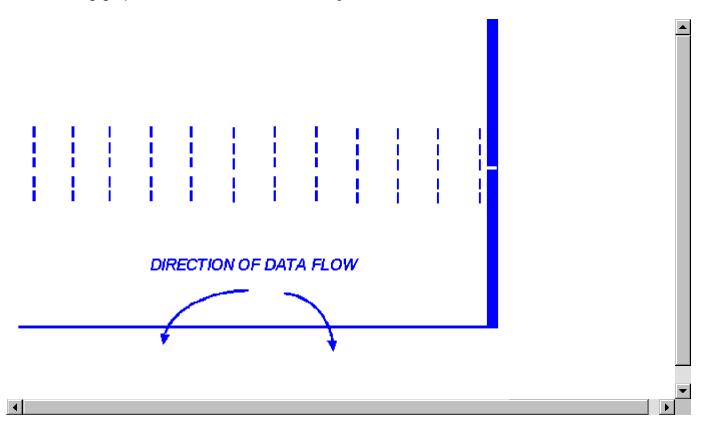
- **Frequency shift keying** (**FSK**): This technique is built on the principle of frequency modulation (FM) of the carrier signal.
- **Phase shift keying** (**PSK**): This technique is built on the method of phase modulation (PM) of the carrier signal.
- **Quadrature amplitude modulation** (**QAM**): This technique involves changing the amplitude and frequency of the carrier wave at the same time.
- **Trellis coding modulation** (**TC** or **TCM**): This is a variant on QAM that includes a method for forward error correction; that is, error correcting information is already carried with the data.

rather than requiring retransmission to correct errors.

Any one of the modulation standards can use the same modulation technique as another standard; however, the two modulation standards still may not be compatible if they do not specify the same amplitudes or frequencies.

ASYNCHRONOUS DATA STREAM

The following graphic illustrates the bits in an asynchronous data stream.

DIRECTION OF DATA FLOW

Each character or byte is mapped or coded onto the carrier signal as a series of pulses. Each character is preceded by a start pulse or start bit, and is followed by a stop pulse or stop bit. Each pulse is typically about 0.0001 seconds long or *wide*. The stop bit is usually 1, 1.5, or 2 *bit times* wide. A data *packet* in its simplest form consists of the start bit, seven or eight data bits, and one or two stop bits. If parity is used there are only 7 data bits. The amount of idle time between the characters varies. Another way of saying this is that packets are spaced *asynchronously* along the carrier wave, giving the protocol its name.

Some of the terminology from the earliest days of the telecommunications industry still survives today. An implementation of the telegraph used a stylus to place a *mark* on a piece of paper. Lifting the stylus resulted in a *space*. Today, when the line is idle (no data mapped onto the carrier wave), we say the voltage level is in a *mark* state. A start bit is a change to the *space* state. The state is altered from a mark to a space or a space to a mark to code the data bits onto the carrier wave. A data bit represented by a space is interpreted as a logical zero (0). A data bit represented by a mark is interpreted as a logical one (1).

Bits are sometimes dropped because of noise. A *parity check* can be added to the data packet to recover from errors in which a single bit is dropped or added. Many communication software packages allow you to configure parity settings, or you may also be able to specify parity information through the serial port configuration options in an operating system such as the Microsoft® Windows® operating system. Parity, the number of stop bits, and the number of data bits, can all be set as serial port parameters.

If a parity bit is used, then an extra bit is added to each character to make the total number of ones (1s) in the character either odd or even, depending on which parity type is used. If you configure your communication software to use parity bits, then there are seven data bits. If parity is not used (set to
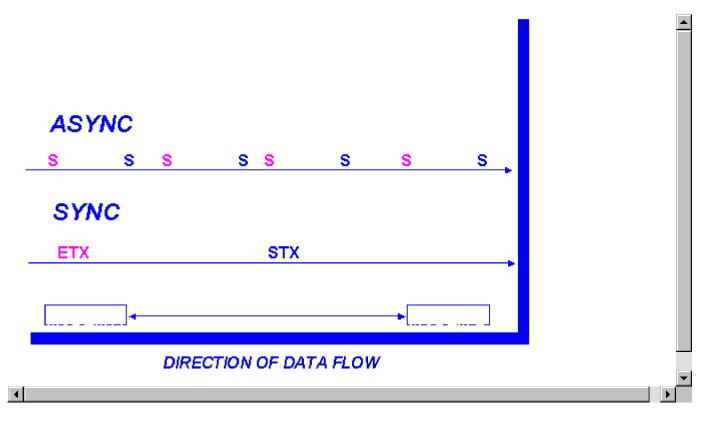
Some parts of the physical cabling in RS-232 dial-up connections are *two-wire;* for example the connection between the computer and the modem. Older modems and modulation standards can only be operated in *half-duplex* transmission mode over physical two-wire connections. Half duplex means that transmission can only happen in one direction at a time. When a modem receiving in half-duplex mode wants to send data, it signals the sending modem. When the sender detects the *turn around* request, it starts receiving and the first modem sends. Not sending and receiving in both directions at once is obviously much slower than full duplex mode, and the line turn around requests delay the data transmission even more.

It is much faster to transmit in both directions simultaneously, in *full-duplex* mode.

The carrier frequency defines the total bandwidth of the connection. Modems can split the carrier frequency into two separate bandwidths or channels to provide two-way full-duplex transmission over a two-wire physical connection.

ASYNCHRONOUS vs. SYNCHRONOUS TRANSMISSION

The following graphic illustrates an asynchronous versus a synchronous data stream.

In asynchronous transmission, data is transmitted one character at a time. Each character is bracketed by a start bit and one or more stop bits. Asynchronous transmission is also called *start-stop* transmission. The asynchronous characters are not evenly spaced along the transmission medium. In gaps between characters, the line is idle; nothing is transmitted. The characters are transmitted independently with regard to timing signals.

In synchronous transmission an extra signal, the clock signal, is added to the RS-232 leads. Usually, the two synchronous modems supply the clock signal, although the DTE (PC) can also supply the clock signal. Configuring a synchronous port for *external clock* means the modem supplies the clock signal. In synchronous transmission, characters are grouped together in large blocks of data, sometimes 2,000 to 4,000 characters to a block. There are no spaces between characters in the block, whereas there are always spaces between asynchronous characters, and therefore, some idle line time. A control character, called a *SYN* character, always precedes the data block. Usually two or three SYN characters precede the data. The block starts with a start-of-text (STX) character. An end-of-text (ETX) character terminates the block.

During a transmission stream, blocks are evenly spaced along the medium with respect to time. If no data is being transmitted, SYN characters are transmitted periodically to keep the two modems *synched up.* Generally, modems can achieve a higher bit rate using synchronous transmission, but such transmission requires more complex and therefore more expensive hardware because of the need for a clocking signal. Also, if you send a large block of data synchronously and an error occurs, there is a greater speed penalty for re-transmitting the large packet, as opposed to just re-transmitting a single corrected packet when using asynchronous transmission. However, in asynchronous transmission, the added need for a start bit and one or more stop bits per character may add 10 to 20 percent to the size of the transmission. Nonetheless, asynchronous modems currently enjoy wider use.
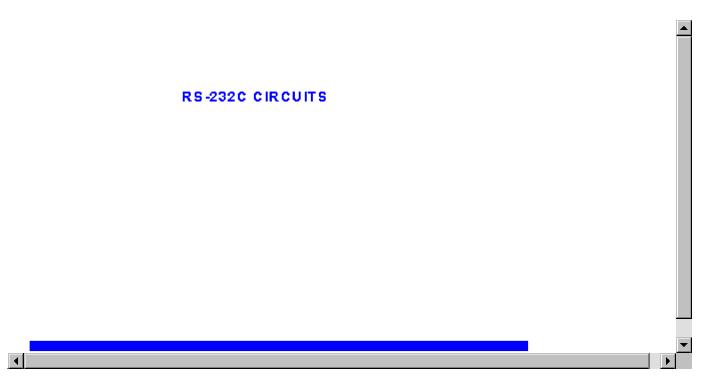
EIA STANDARDS

The EIA issued RS-232-C in 1969. It then issued the RS-449, RS-422, and RS-423 standards in 1970, expecting that they would replace RS-232-C. The RS-4xx standards were designed to support higher modulation rates than RS-232. However, equipment manufacturers continued to use RS-232-C, making it the *de facto* standard. In 1987, the EIA issued the RS-232-D and RS-530 standards.

The EIA standards define the physical-layer circuits and basic signaling. Other standards, such as data compression or error correction standards, are coded in the data-link layer. In its simplest form, asynchronous communication requires just the signals of the physical RS-232 layer and the basic modulation standard. Getting two different modems to communicate sometimes means negotiating down to the RS-232 standard and an older modulation standard.

The RS-232-D standard defines extra test signals between modems, formally specifies the DB-25 pin connector, and defines a *protected ground conductor* for extra shielding from stray radio frequency (RF) signals. In Europe, the ITU-T V.24 standard is the most widely used. (ITU-T stands for International Telecommunications Union - Telecommunications Standards Sector.) ITU-T V.24 is nearly identical to the RS-232-C specification. Therefore, in practical terms, you are not likely to encounter modem devices claiming to conform to the RS-4xx standard. Almost all modem devices and connectors will conform to either RS-232C, RS-232D, or ITU-T V.24 standards, and these three standards can be considered to be equivalent for most *real world* configurations.

ASYNCHRONOUS CIRCUITS AND PIN ASSIGNMENTS

RS-232C CIRCUITS

This diagram shows the signals used in asynchronous communication and the connector pin number the circuits are assigned to by the RS-232C standard. Two types of physical connectors or plugs are used. The left column shows the pin assignments for 9-pin cabling, and in the right column, pin assignments

connectors.

Handshaking between a computer (DTE) and a modem (DCE) is done by raising and lowering voltage levels on these pins. Each of the represented circuits may or may not have a corresponding indicator light on the modem. This light, if present, is known as a light-emitting diode (LED).

RS-232-C defines nine electrical circuits that are used for the protocol handshaking. The following sections describe what the nine circuits do.

Carrier Detect (CD)

When the local modem detects a carrier signal from the remote modem, it raises the voltage level on this lead, indicating to the computer that a carrier signal is being received from the remote modem. This signal is also called the Received Line Signal Detector (RLSD).

When a carrier is detected, the CD indicator on the local modem goes on solid. If the CD LED goes off, the connection has been dropped.

The carrier signal is a tone or frequency. You can hear it usually just after you hear the originating modem dial. It is the sine wave signal that digital data from the two computers is mapped onto by the modem's modulation function. Without a carrier there can be no connection. If this signal is dropped during a connection, your communication software will usually display a *lost carrier message*. If a carrier is not detected at the start of a connection, the calling or originating modem displays *no carrier*.

If the CD indicator is on for a little while and then a carrier is dropped, it probably means the two modems could not negotiate to a common set of standards. This lead is essential to asynchronous communication. If the modem does not have a panel LED to indicate CD, turn on the speaker and listen for a carrier. Clear to Send (CTS) will not be signaled to the serial port if CD is not detected. There should be a CTS LED, especially if there is no indicator for a carrier.

Receive Data (RD)

When the local modem demodulates data received from the remote DCE (modem), the demodulated data is sent to the computer serial port by raising and lowering the voltage level on this lead. The modem LED may be variously labeled as "RD," "Rx," or "RxD," depending on the modem manufacturer. When the RD indicator is blinking rapidly, it means you are receiving data from the remote modem. When RD goes off, the remote modem has stopped sending. This lead is essential to asynchronous communication.

Transmit Data (TD)

The computer sends data to the local modem by changing the voltage level on the *Transmit Data* circuit, pin 3 on 9-pin RS232-C connectors. The Transmit Data LED may be labeled as "TD," "SD," "Tx," or "TxD," depending on the manufacturer. If the computer is sending data, the TD indicator on the modem should be blinking rapidly. At the same time, the RD indicator (LED) on the remote modem should be blinking rapidly to show data is being sent to the remote computer. This lead is essential to asynchronous communication.

Data Terminal Ready (DTR)

The computer turns this circuit on (asserts DTR) when it is ready to be connected to the phone line. If the modem does not get this "ready" signal, no attempt to dial can occur and no commands from the computer will be received. If the modem indicator panel has a DTR LED, it should be on solid before dialing can be attempted. Sometimes the LED is labeled TR (Terminal Ready) instead of DTR.

Signal Ground (SG)

Signal Ground (SG) provides a reference level or benchmark voltage for the other leads (circuits). Asynchronous signals (events, state changes) are sent between the serial port and the modem by

the voltage on pin 4 to the voltage level that indicates "ON". The voltage level, actually a range, is dictated by the RS232-C standard. However, for the modem to make sense of the voltage change, a constant benchmark voltage is needed; this is the voltage level on pin 5, SG, which is considered to be the zero voltage level. There is no LED indicator for SG.

Note that there is another ground, called *protective ground*, which is like the third prong on a three-prong appliance plug. Protective ground is pin 1 in 25-pin connectors, and if it is present, SG is usually strapped or connected to protective ground (chassis ground). This pairing is not possible on a 9-pin connector because the DB-9 does not have a protective ground lead.

Data Set Ready (DSR)

The modem turns this circuit on (asserts DSR to the computer) when it is ready and physically connected to the phone line. If the terminal does not get this ready signal, any attempt to dial will fail. If the modem indicator panel has a DSR LED, it should be on solid. The computer application will not be able to dial if this signal is not present from the modem. Usually, the modem indicator LED is labeled MR (Modem Ready) instead of DSR.

Request To Send (RTS)

The RTS circuit is turned on by the serial port to tell the modem that it actually has data queued for transmission. If the modem has CTS (Clear to Send) turned on, actual transmission will start. CTS off, signaled by the modem to the computer, tells the serial port that the modem temporarily cannot accept any data for transmission. When the asynchronous application writes a buffer destined for the remote application, this causes the serial port to raise (turn on) RTS. RTS and CTS are used for *flow control* between DTE and DCE. (See the Flow Control section of this document for more information.) Usually, there is no LED indicator on the modem front panel for RTS.

Do not confuse Request To Send (RTS) with Send Data (SD, or alternatively TxD, TD, or Tx). SD is always a panel LED and blinks rapidly to show that the modem is actually receiving data from the DTE. RTS is a state preparatory to actually sending to the modem. It is like a query by the DTE, asking if the modem is ready to receive data.

Clear To Send (CTS)

The CTS circuit is turned on by the modem to signal the serial port that the modem wants to write to the computer. If the serial port has Request To Send (RTS) turned on (responds with RTS), the modem will actually transfer data to the serial port. This is, of course, data received from the remote modem. Data is demodulated as it is transferred to the serial port. Data is accumulated byte by byte in a buffer by the serial port. When the buffer is full, it is transferred to an application buffer and is then ready to be filled again by the modem. If the serial port cannot accept data from the modem (DCE), it turns RTS off and the modem waits for RTS on. If RTS is off most of the time, it usually means the application is too slow. The modem is receiving from the remote DCE faster than the local application can read and process data.
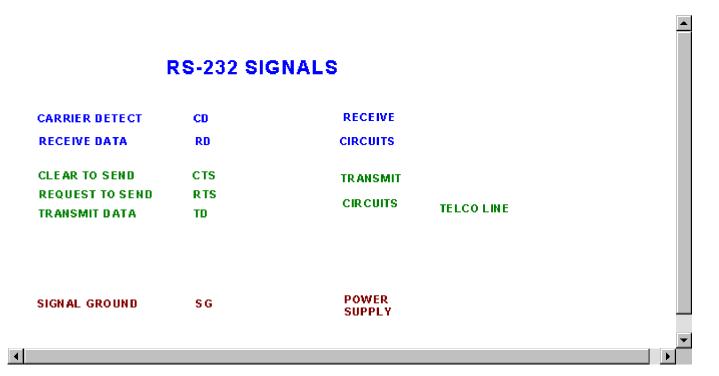
Sometimes there is a CTS indicator on the modem panel. If there is an LED for CTS, it should either be on solid (especially when the local DTE is supposed to be receiving lots of data) or blink occasionally.

Do not confuse Clear To Send (CTS) with Receive Data (RD, or alternatively Rx or RxD). RD is always a panel LED and blinks rapidly to show that the modem is actually writing data to the serial port. CTS is a state preparatory to actually writing to the computer by modem. It is like a query by the modem (DCE), asking if the DTE is ready to receive data.

Ring Indicator (RI)

The RI circuit is turned on by the modem to signal the serial port that there is an inbound call. The line rings for one second, and then there is a four second pause. The RI circuit is turned on for each ring. It serves as a *wake-up* signal to the DTE. The DTE responds by asserting DTR to tell the modem that it is ready to be connected over the telephone company line. If Carrier (CD) is detected, then data transmission starts.

The following graphic illustrates the sequence of events that occurs in RS-232 communication.

RS-232 SIGNALS

CARRIER DETECT	CD	RECEIVE
RECEIVE DATA	RD	CIRCUITS
CLEAR TO SEND	CTS	TRANSMIT
REQUEST TO SEND	RTS	CIRCUITS
TRANSMIT DATA	TD	TELCO LINE
SIGNAL GROUND	SG	POWER SUPPLY

The RTS/CTS and DTR/DSR pins are the main handshaking signals. Pin assignments for both 9-pin and 25-pin connectors are given in the text below. The 9-pin is always stated first. The numbers in the diagram above shows the order or sequence of the signaling when a modem receives a call and data transfer occurs in full-duplex mode. The signals are discussed below in the order of the above sequence.

DSR ▲

When the modem is powered on, it asserts DSR to the serial port by raising the voltage on pin 6. As stated earlier, there usually is a modem LED labeled MR (Modem Ready), which should be on.

RI and DTR ▲

When an incoming call is received, the modem raises the RI voltage (pin 9 or 22). When the serial port detects the RI voltage, it asserts DTR, pin 4 or 20. When the modem raises RI, it must then detect DTR from the computer. The RI modem LED will go on. Sometimes there is a DTR or TR (Terminal Ready) LED on the modem panel. It should light shortly after the RI LED lights. Optionally, the computer can also turn on DTR when it is booted, prior to receiving the RI signal.

CD ▲

The calling modem transmits a carrier tone to the answering modem. When the answering modem detects the carrier tone, it knows the circuit to the remote modem is complete. The answering modem signals its presence to the computer by raising the voltage on pin 1 or 8. When the serial port detects RI and CD, it knows there is an incoming call.

RTS ▲

To send data, the computer asserts RTS by raising the voltage on pin 7 or 4. Even though in this example the local DTE did not initiate the call, it must first signal the remote DTE that it is ready to receive data. At this point, the computer application will usually transmit a logon or hello message to the originating end (Held, page 88). Raising RTS on the receiving DTE will result in turning on CD at the remote or calling modem (McNamara, page 57). This will occur as soon as CTS gives the "OK."

CTS

The modem responds to RTS by asserting CTS. It does this by raising the voltage on pin 8 or 5. When the computer detects CTS, it knows that the modem is ready to send the *ready to receive* message to the remote DTE.
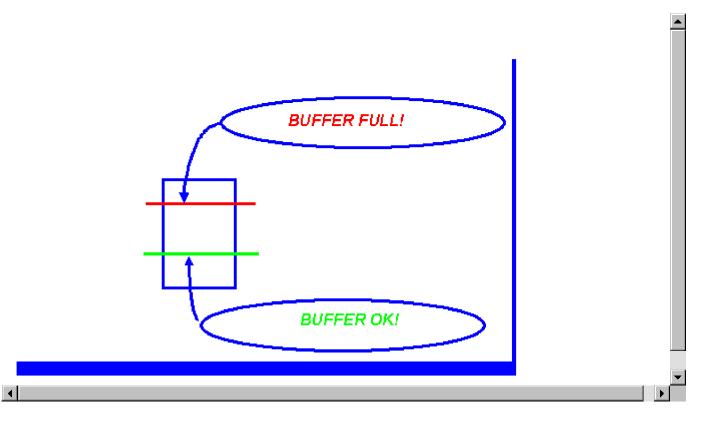
RD and TD

At this time, the computer will receive data on the RD pin (pin 2 or 3) and transmit data on the TD pin (pin 3 or 2). The modem TD and RD LEDs should blink rapidly.

Notice that the perceived order is different for the sending modem, and the exact handshaking sequence for your modem may vary slightly from the sequence presented here. The sequence will vary between a half-duplex and full-duplex transmission. Also, some latitude exists in the RS-232 standard for specific implementation in both the hardware and communication software (McNamara, pages 53, 57, 80-86).

DATA FLOW

Generally the rate of data transfer from the computer through the serial port to the modem (known as the DTE rate or the local transmission rate) is much faster than the transmission speed over the modem-to-modem connection (referred to as the link rate). This is because current versions of the UART chip are capable of sending serial data to the modem at a rate of 115.2 Kbps, while the typical modem transfer rate is much less than that. With this kind of difference in data transfer rates, the potential exists that the computer will overwhelm the modem with data (sometimes referred to as *data overrun*), causing either lost data or the need for retransmissions. However, the reverse of this situation can also occur if a newer high-speed modem is connected with an older computer that has an earlier version of the UART chip installed. Some of the earlier UARTs were only capable of a maximum speed of 28.8 Kbps or slower. Therefore, it is important that your modem is capable of handling the speed of the data transfer from the serial port of the computer and vice versa.

One of the ways that computers and modems take care of differences between data transfer rates is through the use of buffers. Both the communication application on the DTE side and the modem on the DCE side can use buffers as a temporary storage location for data. However, problems can still occur even with buffers in place.

When the asynchronous application does a write, the bytes are queued in one or more *buffers*. The

serial port to a buffer in the modem. The asynchronous application in a single write can fill a large buffer; for example, 1024 bytes. Because the modem sends or transmits data one byte at a time over the line, the modem buffer can be overwhelmed. Application data in the modem buffer can be overwritten and lost. When the application is reading data, the modem can transmit too fast, causing data in the application buffers to be overwritten. To keep this from happening, for data going in either direction, *flow control* is implemented.

When the modem's buffer is so full that additional data written into the buffer causes data to be overwritten and lost, the modem has to signal the serial port to stop sending data. For inbound data, if the application can't process the data fast enough (that is, it empties the modem buffer too slowly), then the serial port has to signal the modem to stop sending. This handshaking or signaling is called flow control. Typically, flow control will be implemented in one of three ways: RTS/CTS or hardware flow control; XON/XOFF or software flow control; or ENQ/ACK or Enquire / acknowledge flow control.

RTS/CTS or Hardware Flow Control ▲

With RTS/CTS flow control, when the serial port wants to send data to the modem, it signals Request To Send (RTS) to the modem. If the modem can accept the data (that is, the buffer not too full), it returns a Clear To Send (CTS) to the serial port. If the modem cannot accept the data, it turns off the CTS signal. On a 9-pin connector, RTS is signaled by raising or lowering voltage levels on pin 7; CTS is signaled by raising or lowering the voltage on pin 8. For example, if the modem cannot receive any more data, it drops the voltage on pin 8, turning off CTS. Sometimes the application cannot process its buffers fast enough. To keep the modem from over-writing data, the application can tell the serial port to drop RTS (lower the voltage level on pin 7). The DCE (modem) senses that RTS is dropped and stops filling the application buffer. When the application has emptied some of its buffers, it raises RTS again and the modem starts filling the buffer.

XON/XOFF or Software Flow Control ▲

When a modem is receiving data from the local terminal or serial port too fast for the modem to process the data, the modem will send an ASCII 19 (CTRL+S) or DC3 (device control 3) character to the serial port. This is the default XOFF character. It is sent on pin 3 (RD).

When the modem buffer is no longer full (that is, data has been transmitted to the remote modem), the modem sends the XON character—an ASCII 17 (CTRL+Q) or DC1 character to the serial port, and the serial port starts sending data again. The characters used for XON/XOFF can often be changed in the application and the modem, if desired.

One problem with XON/XOFF is that the CTRL+S and CTRL+Q characters can be characters embedded in a data file. The data is interpreted as DC1 and DC3 characters, causing transmission errors. Modems and DTEs using XON/XOFF flow control do not need all nine RS-232C circuits or pins shown in the RS-232 Signals diagram in this document. XON/XOFF can, and often does, operate with just three pin-outs, CD, RD, and TD, although it is advisable to also have at least SG. A cable that works with an XON/XOFF application may not work with an application that uses hardware flow control, because applications that use hardware flow control need all nine signals or pins.

ENQ/ACK or Enquire/Acknowledge Flow Control ▲

In ENQ/ACK flow control, when the serial port wants to send data, it transmits an ASCII ENQ (enquire) character. If the modem can accept data, it ACKs the serial port (transmits back the ACK, *positive acknowledgment* character). If it can't accept the data (buffer full), the device or modem sends back a NAK, or *negative acknowledgment*, in response to the ENQ. ENQ/ACK flow control is used in some types of Hewlett-Packard computers and devices.

ITU-T AND PROPRIETARY STANDARDS ▲

The Consultative Committee on International Telegraphy and Telephony (CCITT) was responsible for first proposing international standards for data communication over telephone lines using modems. Their standards begin with the letter "V" and are therefore known as the "V series" standards. Sometimes a

given standard may be followed by the word *bis* which is French for "second." This indicates the first revision of the original standard. The word *ter*, or "third," is used to represent the second revision of a particular standard. The CCITT eventually changed its name and is now known as the International Telecommunications Union - Telecommunications Standards Sector (ITU-T).

In addition to the official international standards, some proprietary standards have become widely accepted by the industry. A good example of such a standard created by a modem manufacturer is the Microcom Networking Protocol (MNP). MNP defines ten levels of data compression and error correction schemes. MNP begins with the simplest level, Level 1, and progresses to the most advanced level, currently Level 10. MNP Level 1 is obsolete and no longer implemented by manufacturers. Level 8 was never implemented. Levels 1 through 4 are now part of the public domain, and modem manufacturers can license most of the other levels for use.

In the next section, some of the most important and widely used standards for error correction, compression, and modulation are discussed.

Before 1989, there was no standard for error correction and detection. Several vendors had developed sophisticated error correction and/or detection schemes based on cyclic redundancy checks.

By 1989, the Microcom algorithms had become so widely used that MNP was the *de facto* standard.

When the ITU-T issued the V.42 error correction/detection standard in 1989, it made MNP an optional standard within V.42. The ITU-T V.42 error correction/detection standard is called LAP-M (Link Access Protocol-Modem).

Cyclic Redundancy Check (CRC)

When the first modem has accumulated a buffer full of data for transmission, it applies a formula to the data block, calculating a value D. A second formula is applied to the data and the result is divided into the first value. The final result is a whole number quotient and a remainder.

> *Apply 1st formula to data, result = D*
> *Apply 2nd formula to data, result = G*
> *Calculate D/G = Q + R*

D is the result of applying the first formula to the data block. G is the value of the second formula. The remainder, R, is the cyclic redundancy check (CRC). It is usually 16 bits long and is appended to the end of the data block when the modem, which is not sending at that time, transmits.

When the receiving modem gets the data block, it goes through the same calculation. If there was a transmission error, the second CRC value will not equal the original one, and the receiving modem transmits a negative acknowledgment (NAK). The originating modem then re-transmits the data block. If the two CRCs match, the receiving modem transmits a positive acknowledgment (ACK). The most commonly used formula, or polynomial, is called CRC-16.

Microcom Networking Protocol (MNP)

MNP modems connect first with a lower class protocol, and then negotiate up to the highest class. The higher the class, the greater the throughput. If the connection is to a non-MNP modem, the MNP modem will negotiate down to a *dumb* asynchronous standard, probably basic RS-232-C.

Please understand that *error correction* as it is commonly implemented by industry really means re-transmitting good copies of data packets that were corrupted in the original transmission. Retransmission of packets takes time. Therefore, in general, error correction slows data transmission rates.

Error control is the term used to reference a method for reducing retransmissions. Because MNP levels include other speed optimization schemes besides error control, the improved efficiencies of speed mentioned in the following paragraphs are only partially the result of implementing error control.

the baud rate. For example, a 2400 bps modem running MNP Class 2 will only operate at 2000 bps, or at approximately 85 percent of efficiency. Class 3 is a *synchronous* protocol that gives about 115 percent efficiency. Under Class 3, the two modems communicate synchronously, but the terminal device (DTE) is asynchronous. MNP does not offer much performance benefit until at least Class 4 operation is achieved.

MNP error correction, using the CRC algorithm, actually starts in Class 4. Modems that can use MNP error control advertise or document themselves as being "MNP Class 4 Compliant." Besides error control, MNP Class 4 uses two performance enhancing techniques, Adaptive Packet Assembly (APA) and Data Phase Optimization (DPO), both of which improve throughput.

APA: If the phone connection is error free (very few CRC errors), then MNP assembles larger data-packets for transmission. If errors start occurring, the packet size gets smaller.

DPO: Data Phase Optimization eliminates administrative overhead in packet transmission if the line is error free.

A Class 4 modem gives 120 percent efficiency, so that a 2400 bps modem running Class 4 would appear to operate at 2900 bps.

ITU-T V.42—Link Access Protocol-Modem (LAP-M)

V.42 is the ITU-T recommendation for error detection and correction for asynchronous transmission. Within V.42, LAP-M is the primary standard; however, because MNP was in such widespread use, the ITU included MNP as an option under the V.42 error correction and detection specification. This means that some V.42 modems support MNP Class-4 and some do not. Most V.42 modems incorporate up to MNP Class 4.

At the start of the handshake, the originating LAP-M modem transmits a bit pattern called the originator detection pattern (ODP). The answering modem responds with a bit pattern called the answer detection pattern (ADP). The ADP format specifies that V.42 is supported or that no error correction will be used. If V.42 (LAP-M) is not available, sometimes MNP is negotiated or the modems operate with no error correction.

LAP-M uses a different CRC formula and has a different header than MNP. This means MNP and LAP-M are not compatible.

This section contains three widely used methods of data compression, examples of how this technology works. The first and simplest method is known as Run Length Encoding (RLE). RLE counts repeated characters or bit patterns, and then replaces the repeated characters with a representative bit pattern and a multiplier equal to the number of times the pattern or character is repeated.

The Lemple-Ziv-Welch (LZW) algorithm inspects the data for recurring sequences of information. LZW then builds a dictionary of repeated sequences. A pointer to the appropriate dictionary entry can now replace each repeated sequence in the data stream.

Huffman encoding looks at the data and counts which characters are most frequently used. These characters are assigned the shortest bit values. At the same time, the least frequently occurring characters are assigned the longest bit values.

Standards for data compression have about the same history as for error correction. By 1989 or 1990, the compression algorithms in Microcom's MNP had become a *de facto* standard, partly because Microcom licensed MNP widely to other vendors. By 1990, 100 or so vendors were incorporating MNP compression in their products. In 1990, the ITU-T issued its own compression standard, called V.42*bis*; it did not recognize MNP compression as an alternative, as it did with error correction.

You may want to turn off data compression if you are trying to send a previously compressed file such as a ZIP file. Recompressing an already compressed file may actually increase the total packaged file size. This increase will diminish the speed benefit gained from the initial compression.

MNP Class 5

MNP Class 5 adds data compression to MNP Class 4. MNP Class 5 uses the Run Length Encoding (RLE) algorithm for data compression at its first level. However, MNP Class 5 data compression has a second level, called adaptive frequency encoding, that is applied after the initial run-length encoding. In *adaptive frequency encoding*, a token is substituted for the three repeated data bytes. An attempt is made to use the smallest tokens so that less than 8 bits are needed for each substitution. The smaller ASCII values are continuously changed or adapted to represent the replaced characters. The token consists of three bits of header and the body, which contains the bits representing the repeated byte. Compression increases efficiency by 1.3 to 2 times. MNP Class 5 compression will typically increase efficiency by 1 to 1.6 times, giving a 200 percent increase in efficiency, so that a 2400bps modem will give a 4800 bps throughput.

MNP—Class 7 Enhanced Data Compression ▲

MNP Class 7 data compression starts with run length encoding; however, as the transmission continues, MNP Class 7 builds the tables needed to enable Huffman encoding.

Class 7 compression gains about 40 percent efficiency, which is added to Class 5 compression efficiencies. An MNP Class 7 modem can negotiate down to Class 5. Class 5 predates Class 7 and is more popular.

V.42bis—ITU-T Data Compression Recommendation ▲

V.42*bis* is the ITU-T recommendation for data compression, issued in 1990. V. 42*bis* uses the Limpel-Ziv-Welch compression algorithm.

V.42*bis* compression matches a string in the data stream to a string stored in a series of tables (string dictionaries) and substitutes a code word for the string. Then the code word is transmitted. V.42*bis* is about 10 percent more efficient than MNP Class 7.

This section lists some of today's most important modulation standards. The easiest way to classify a modulation standard is by the data transfer rate, or speed, measured in bits per second (bps). The early standards covered data transfer rates of about 300 bps to 4800 bps. Most of the low and medium speed standards (300 bps to 4800 bps) are either Bell standards or CCITT standards. Most of these standards are now obsolete, although some will be included in a new modem's capabilities as a *fallback* standard for backward compatibility with older technology. Note that when two modems agree to connect with a specific standard, it is still possible for them to dynamically negotiate certain parameters within that standard. Microcom modems, for example, can adjust for line noise by changing the negotiated frequency on the fly.

Currently the most widely implemented and used modulation standards cover data transfer rates from 9600 bps to about 33,600 bps. V.32, V.32*bis* , V.34, and V.34-1996 are the ITU-T high-speed asynchronous standards. There are also proprietary high-speed modulation standards (9600 bps and faster), and there is a currently emerging standard for a speed of 56,000 bps.

Sometimes two different modulation standards use the same physical modulation technique, but are still not compatible because some other part of the standard is different. For example, two standards may use the same modulation technique, but use different frequencies within the carrier band, making them incompatible.

Even adherence to the same standard by two modem manufacturers does not necessarily mean that the two modems will be able to communicate successfully. Adherence to a standard simply means that the standard is not violated. And even if the standard is not violated, one manufacturer may follow only one portion of the standard's requirements, while the other manufacturer may follow a different part of the requirements for the same standard.

Finally, emerging standards can be developed around new technologies, but different manufacturers may implement the new technology in a different way, causing temporary incompatibilities until the *de facto* standards become formalized into internationally accepted specifications. A current example of this is the technology facilitating 56,000 bps speeds. Two current implementations of 56 Kbps technology, the X2

and 56K Flex modems, are both proprietary and incompatible with each other. At the same time, a new international standard for 56 Kbps, the V.90 standard, is currently being refined by the ITU-T. It may be quite some time before the leaders in 56 Kbps technologies produce modems that truly adhere to the same V.90 standard and can also communicate with each other.

ITU-T V.32—9600 Bps Standard

ITU-T's V.32 standard was issued in 1989 for asynchronous, full-duplex operation at 9600 bps.

Although designed for asynchronous DTEs, two V.32 modems actually communicate synchronously. A circuit converts the asynchronous data stream into synchronous blocks, invisible to the application. V.32 supports modulation rates of 2400, 4800, and 9600 bps. Data compression and error correction can increase the throughput rates to 14.4, 19.2, and faster.

ITU-T V.32bis—14.4 Kbps Standard

ITU-T's V.32 standard was issued in 1991 for asynchronous, full-duplex operation at 14.4 Kbps. V.32bis is an extension of the V.32 technology. V.32*bis* supports modulation rates of 2400, 4800, 9600 bps and 14.4 Kbps. Data compression and error correction can increase the throughput rates.

V.FC

This was an interim modem standard designed to support speeds up to 28,800 bps. V.FC stands for Version Fast Class. This standard is proprietary and is not compatible with the ITU-T V. 34 at speeds faster than 14.4 Kbps.

ITU-T V.33—14.4 Kbps Standard

ITU-T's V.33 standard was issued in 1991 for synchronous, full-duplex operation at 9600 bps and 14,400 bps. Although V.33 was designed for synchronous DTEs, conversion circuits can be added. The conversion of the asynchronous data stream into synchronous blocks is invisible to the application. V.33 is an extension of V.32 modulation techniques. V.33 supports data rates of 2400, 4800, 9600, and 14,400 bps, but proprietary enhancements by modem vendors, mainly addition of data compression, allows for faster V.33 rates.

ITU-T V.34—28.8 Kbps Standard

By September 1994, ITU-T's V.34 standard was widely adopted over the proprietary 28.8 Kbps standard V.FC (V.Fast Class). V.34 has been more widely adopted than any other 28.8 Kbps standard.

ITU-T V.34—1996

This is an emerging standard for modem speeds of 33.6 Kbps. While being formulated, the standard has also been known as the v.34*bis* or the v.34+ standard. The v.34-1996 standard modem is more robust than modems implementing the v.34bis or v.34+ interim standards. With the v.34-1996 standard modem, users are much more likely to experience a 28.8 Kbps connection on a noisy line with the hope of attaining possible speeds of 31.2 Kbps or 33.6 Kbps under optimum conditions.

For two modems to communicate, they must first find a common modulation standard. If two modems connect briefly and then time-out (that is, the CD LED comes on for several seconds then goes off), it is because they have failed to find a common modulation standard. The first thing the two modems negotiate is the modulation standard; however, some modem manuals call this "negotiating the baud rate," obscuring the fact that there is always the possibility of a modulation standards compatibility issue. Check the modulation standards supported in the two modems' reference guides. You may be able to force a common standard if the default negotiation fails. However, the two modems must have a common modulation standard, such as V.32 or V.34.

Proprietary modulation standards often offer faster data rates than public standards, such as Telebit PEP

However, a connection using a proprietary modulation standard can only be achieved if the proprietary standard is available in the modems on both ends of the connection. If the two modems are from different vendors or are different models, the best strategy for connecting is to look for a common ITU-T standard.

Most modems will attempt to negotiate down from the highest modulation standard they are capable of to the lowest modulation standard that they are capable of. The following are some of the modulation standards of *last resort*. Almost all modems support V.22 (1200 bps) and V.22*bis* (2400 bps). If MNP Class-5 data compression can be added to the V.22 or V.22*bis* standards, acceptable throughput may be achieved. For example, Microcom QX series achieves 19.2 kbps with V.22*bis* and MNP Class 4 and 5.

Bell System standards such as 103, 212, and 202(x) ranging from 300 to 1800 bps (unless enhanced) are too slow to be used for anything other than a temporary connection solution. For example, connecting a Telebit modem to a US Robotics modem using these standards allows you to connect temporarily. Note that Bell System standards are not used in European markets.

Anderson, Douglas, Patrick Dawson, and Michael Tribble. *The Modem Technical Guide*. Boulder, CO: Microhouse International, 1996.

Black, Uyless D. *Data Link Protocols*. Englewood Cliffs, NJ: Prentice Hall, 1993.

Dvorak, John C., Nick Anis, and Werner Feibel. *Dvorak's Guide to PC Connectivity*. New York: Bantam Books, 1992.

Held, Gilbert. *The Complete Modem Reference*, 2nd ed. New York: John Wiley & Sons, 1994.

Lewart, Cass R. *The Ultimate Modem Handbook*. Upper Saddle River, NJ: Prentice-Hall PTR, 1998.

McNamara, John E*. Technical Aspects of Data Communication*, 3rd ed. Digital Press, 1998.

Newton, Harry. *Newton's Telecom Dictionary*, 9th ed. New York: Flatiron Publishing, 1995.

Richard, Jack. "The 56K Modem Battle." *Boardwatch Magazine* < www.boardwatch.com >. Vol. XI: Issue 3. March, 1998.

Seyer, Martin D. *RS-232 Made Easy: Connecting Computers, Printers, Terminals, and Modems*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

Contributor - Eric Bernstein, *EBCS, Computer Service Engineer*, ericber@oz.net.