Introduction to Modems

2004 September 21 by Anthony Glenn
File: Introduction to Modems.sxw

Many modem users feel they know too little about things like com ports, UARTs, port speed, carrier speed and the like. They wonder why optimising a modem's performance seems so hard. The purpose of this document is to provide some information on these subjects. Any modem commands in this document are for modems based on a Conexant chipset.

Why Modems are Needed

Sending computer data over long distances is not so simple. A telephone line is the only commonly available transmission medium, apart from posting a floppy disk. Ordinary telephone lines are designed for voice. They have a restricted frequency response and various other performance limitations. Trying to send raw computer data straight out of a computer port down a telephone line would not work. Something is needed which will produce a signal capable of being satisfactorily carried by a telephone line and also convert an incoming signal back to computer data. That something is a modem.

The word “Modem” is short for Modulator-Demodulator. A modem takes digital data and uses it to modulate a carrier, the modulated carrier is sent through a transmission medium, then at the far end another modem demodulates the carrier and recovers the original digital data. The carrier is usually a low frequency audio tone. The transmission medium is usually an analog telephone line, as used for normal telephone conversations. From the point of view of the telephone network, a modem connection is just another conversation, it just happens to be modem noises rather than human language.

Computer and Modem

The modem is Data Communications Equipment (DCE). The DCE translates between digital data, from some digital device, and a signal which is suitable for transmission. In this document, the term “modem” is generally used in preference to “DCE”.

The Data Terminal Equipment (DTE) is the thing which sends and receives digital data through the modem. A DTE is usually a computer, but could be a dumb terminal or a printer. In this document, “computer” means a Windows-compatible computer from any manufacturer. Other computers (for example, Macintosh, Amiga, Sun) typically have different hardware, but the general principles remain the same. This document uses the term “computer” instead of “DTE”. If your DTE is actually a dumb terminal or something else, just make the appropriate substitution.

Parts of the computer which affect the performance of a modem are the com port, the CPU, and the software.

Both the computer and the modem have a com port in them, which communicate through a standard interface. The interface is the com port connector. The pin numbers, signal meanings and electrical details of the interface are standardised in an Electronic Industries Association (EIA) standard known as RS-232C.

Connect in Three Steps

Most users will be originating a call from their local modem to a remote modem. What the software has to do to get a data connection between the two modems is a simple three-step process.

1. Set the port parameters on the computer. For example, 115200 bit/s, no parity, 8 data bits, 1 stop bit, RTS/CTS hardware flow control, hang up by turning DTR off.
2. Send a reasonable initialisation string to the modem, and get the OK response back from the modem.
3. Send a correct dial string to the modem, then remain silent. The modem will dial and get connected to the other modem. Get back the connection report from the modem.

The modems are then in data mode. They are transparent, any bytes sent into one appear at the other. There are a few software writers who find it hard to follow this simple recipe. It helps to remember the underlying three steps when troubleshooting modem-related software.

Data Mode Functions

Inside a modem is circuitry to do modulation and demodulation (known as the "data pump"), error correction and data compression. Starting from the computer, there is the com port (part of the computer), the data cable, another com port (built into the modem and not normally mentioned), the data compression, the error correction, the data pump, the DAA (Direct Access Arrangement, that is, the telephone system interface), then finally the telephone line. All these functions have a TX path and an RX path, data is flowing in both directions simultaneously. When two modems are connected, the TX data in one modem becomes the RX data in the other one.

The data compression (V.42bis or MNP5) is between the computer and the error correction. For incoming data from the computer, the data compression looks for repetitive patterns and replaces them with tokens involving less bits. The tokens are expanded again to the original data at the receiving end. The result of this is to gain an effective increase in throughput for repetitive data. The data compression is entirely done within the modems and is totally transparent to the computers, except for the increase in performance.

The error correction (V.42 or MNP2 to 4) is between the data compression and the data pump. It translates all the bytes from/to the data compression to/from "packets" of data which are sent to/from the data pump. Each packet has a cyclical redundancy check word (CRC) which is computed from the data in the packet by a well standardised algorithm. The CRC is recomputed at the receiving end and tested against the CRC sent in the packet. If the test passes, the data is passed through to the data compression. If the test fails, the packet is thrown away and a request that it be retransmitted is sent back to the other modem. So any noise bursts on the telephone line, just result in a few packets being corrupted, which then get thrown away and retransmitted. The computer never sees any bad data from the noise burst, just a tiny hesitation in the flow of data. All this error correction is also being done entirely in the modem, the computer knows nothing about it. The computer is also spared from having to cope with any bad data caused by the telephone line.

The data pump (V.90, V.34, V.32bis, etc.) is between the error correction and the telephone line. The modulation gets raw data (the bits which the packets are made of) from the error correction, then converts it into analog voltages on the telephone line. At the receiving end, the demodulation extracts the analog voltages (sent from the other modem) and converts them back to raw data. Alas, the raw data now has errors in it, depending on how faithfully the analog voltages were transmitted over the telephone line.

Cause of Poor Performance

However, suppose the signal quality out on the telephone line is bad, so there is an excessive number of errors in the raw data. Many packets will be corrupted and thrown away. The modem could spend most of its time throwing away bad packets and only a small amount of time transferring useful data. Throughput would be terrible. Faster carrier speeds are more sensitive to defects in the telephone line than slower carrier speeds. It is easier to demodulate correct raw data at a lower carrier speed compared to a higher. So if a modem is connected at a carrier speed which is too high, the signal quality would be bad. This is how a new fast modem could be slower than an old modem which is only capable of a slower maximum carrier speed.

The answer is to reduce the carrier speed. Lower carrier speed will give better signal quality, less errors in the raw data and less packets thrown away, resulting in better throughput. For a given quality of telephone connection, there is an optimum carrier speed where throughput is maximised. The modem is not good at selecting the optimum carrier speed. The data pump decides for itself whether to go up or down in carrier speed, based on the size of error vectors detected during demodulation. The other modem may not permit changes in carrier speed. There is no linkage from the error correction back to the data pump, such that if the percentage of bad packets goes high, the carrier speed is forced down. The modem can sit there with the data pump thinking, "Signal quality is not too bad, stay at this carrier speed." Meanwhile, the error correction is throwing away a large percentage of packets, so the throughput of useful data is bad. Modems are biased to try to get the highest possible carrier speed, because that usually gets the best throughput, but sometimes a high carrier speed is not the best. That means that you the human being must help the modem by telling it what maximum carrier speed it may connect at.

This problem of the carrier speed being too high could happen on TX or RX or both. The standard rule is: If trouble, go DOWN in carrier speed.

A +MS command must be sent to the modem before it connects, to control the carrier speed. That can be done in various ways depending on the software.

Port Speed and Carrier Speed

With a modem there are three speeds involved, the *port speed*, the *receive carrier speed* (RX), and the *transmit carrier speed* (TX). The port speed is the speed in bits per second between the computer and the modem, through the com port. That speed is usually left permanently at 115200 bit/s. Port speed is often known as “Baud rate”. The carrier speeds are the speeds over the telephone line, RX is from the telephone line into the modem, TX is from the modem out over the telephone line. For a 56k modem, RX is typically somewhere near 44000 bit/s. TX is typically somewhere near 28800 bit/s. It is important to distinguish between these speeds. When you set the speed in software, what you are setting is the port speed. The +MS command is used to control the carrier speeds.

The port speed being set to the maximum is not a problem provided there are negligible occurrences of overrun errors. The problem of overrun errors has largely disappeared, these days. Port speed is usually set significantly higher than RX so that the port speed does not become any kind of limiting factor for performance.

The port speed is given in bits per second (bit/s). Bits per second is the same as the baud rate for digital data. The modem does Hayes-compatible automatic port speed detection on every AT command sent to the modem. In a “Hayes-compatible” modem all commands begin with the characters “AT”. Hayes was a modem manufacturer which invented the early versions of the AT command set. The modem determines the port speed by timing the bit pattern in the characters “AT”. The modem port speed detection ensures that the modem has the same port speed as the computer.

The modem will store the port speed of the last &W command in non-volatile memory. The next time the modem is powered up, that is the port speed the modem will start at. Sometimes a modem which is used for auto-answer does not have commands sent to it between being powered up and getting connected.

The carrier speed is what determines the maximum possible speed at which useful data may be transferred.

All higher speed modems have internal buffering, which allows the port speed to be quite independent of the carrier speed. The port speed remains constant, but the carrier speed may vary from connection to connection, or during a connection.

Some older “direct mode” modems had the restriction that the port speed had to be the same as the carrier speed. Under those circumstances, the software had to be able to change the computer port speed to match the carrier speed reported by the modem when it connected. This feature is known as “auto baud rate” or some similar wording. Make sure that “auto baud rate” is turned off for any modem which does not specifically require it.

There are some carrier speeds which are not permissible as port speeds. Most computers can offer the following port speeds: 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200. Due to the internal buffering which exists in a high speed modem, carrier speeds are often different from any permissible port speed. For example, a V.34 modem can connect at a carrier speed of 26400bit/s, but cannot use that speed on the port. Use a permissible port speed higher than the carrier speed. Flow control automatically prevents lost characters due to the speed difference.

Flow Control

Whenever data is being transferred from one device to another there is always the possibility that the receiving device may not be able to accept data as fast as the sending device can send it, due to the receiving buffer getting full. For example, suppose a computer is sending data to a modem and the port speed is higher than the carrier speed (as would usually be the case), the modem's internal buffer would fill up and the modem needs some way to say to the computer "My buffer is full, stop sending". Likewise, the computer may be slow clearing its own buffer, so it needs some way to tell the modem to stop sending.

This is the function of flow control. Flow control is absolutely universally required. It is needed between computers and printers, between computers and modems, between two modems, even between two software modules within a computer. It is part of the transfer every time data transfer happens between independent processes. It is a low-level, basic function. Flow control must be got right every time, or data errors are inevitable.

The standard method of flow control between computers and modems is RTS/CTS hardware flow control, set by the command &K3 (default). TX (Transmit data, Pin 2) is the pin where serial data flows from the computer to the modem. It is an output on the computer side and an input on the modem side. CTS (Clear To Send, Pin 5) is the flow control line for TX. It is an output on the modem side and an input on the computer side. When CTS is TRUE (defined as being more positive than 3 Volts), data may flow through TX. When CTS is FALSE (defined as being more negative than -3 Volts) data must stop flowing through TX. Data does not have to stop instantly when CTS goes FALSE, a few characters overrun is OK.

What happens is this:

1. TX is sending characters to the modem. CTS is TRUE.
2. The modem receives these characters, but they are coming in faster than the modem can process them. The modem's buffer gets near full. The modem makes CTS FALSE.
3. The computer is checking the state of CTS and sees that CTS is FALSE. The computer stops loading new characters into its com port for sending to the modem. Any remaining characters in the com port chip are sent out on TX. Then the flow of characters stops.
4. The modem receives these few extra characters satisfactorily, because its buffer was not absolutely full when it made CTS FALSE.
5. After a while, the modem has processed the characters which were in its buffer. The buffer gets to be close to empty. The modem makes CTS TRUE.
6. The computer sees CTS TRUE and resumes loading new characters into its com port chip. The new characters start being sent on TX.
7. Go to Step 1 in this description.

TX and CTS work together to ensure that data is transferred from computer to modem without any errors due to buffer overflow.

Similarly, RX (Received data, Pin 3) carries data from the modem to the computer, and RTS (Request To Send, Pin 4) is the flow control line.

Many modems have an RTS light, which is lit when RTS is TRUE. Looking at this light will tell you if your computer buffer is getting full. Similarly for the CTS light and the modem buffer.

CTS has always been the flow control line for TX. Alas, the original meaning of RTS was to tell the modem that there was data to be sent, and for the modem to turn on its carrier. Thus, the meaning of RTS has changed, but the name has not.

There are a few devices which use DTR/DSR as the flow control lines. These devices can be used with a modem provided a custom cable is made. DTR (Data Terminal Ready, Pin 20) normally indicates that the computer is ready to transfer data. DTR can also be used to cause the modem to hang up, when &D2 is set. DSR (Data Set Ready, Pin 6) normally indicates that the modem is ready for data or commands.

Software flow control is also possible. XON/XOFF is the most common software flow control technique between computers and modems, set with &K4. The XOFF character (Ctrl-S) is sent to stop the flow in the opposite direction. The XON character (Ctrl-Q) resumes the flow. Alas, the standard XON/XOFF is not transparent and can be caused to go wrong by XON or XOFF in the data stream. There is a transparent version (&K5), but hardly anyone implements it.

Hardware flow control is transparent and so simple that everyone can implement it correctly. That is why it is so popular. It is the factory default on modems.

Error Correction

All high speed modems support some form of error correction. This is necessary because higher speed modulation techniques are inherently more sensitive to noise on the telephone line. It is unwise to try to connect at a carrier speed of higher than 2400 bit/s without error correction turned on. What typically happens then is that a large amount of spurious characters appears on the RX line, which spoils the appearance of the screen, or upsets the operation of the software.

Apart from getting rid of spurious characters and some change to throughput, error correction is invisible to the computer. The computer does not have to know anything about the error correction protocol built into the modems. You can regard the two connected modems as magic boxes, any data sent to one appears at the other without error. Modem to modem flow control is also achieved automatically. If the far modem gets a full buffer, due to the far computer not accepting characters, the flow of data stops between the modems, then the flow of data stops between the near modem and near computer.

There are different error correcting protocols being used. The most common is known as LAPM (Link Access Procedure for Modems), defined in the V.42 standard from the ITU. Earlier standards were developed by Microcom Inc., they are known as MNP 2 (Microcom Networking Protocol, Class 2) to MNP 4.

An error correcting modem automatically probes for another error correcting modem just after it connects to another modem. Sometimes, an older style answering modem cannot cope with the probe and hangs up. There are modem commands to selectively turn off LAPM or MNP error correction, to avoid this problem.

When an 8-bit byte is sent by a com port, the normal format has 1 start bit, 8 data bits and 1 stop bit. This gives 10 bits on the serial line to transfer 8 bits of useful data. The start and stop bits are needed for clock recovery. It is possible to dispense with the start and stop bits by sending data in the form of packets containing multiple characters. This is done internally by V.42 LAPM. There is some overhead, so a useful rule of thumb is to say that there are 9 bits sent for every 8-bit byte of useful data. For example, say the carrier speed was 14400 bits per second and LAPM was operating, the expected throughput would be 1600 bytes per second. This is better than the 1440 bytes per second you would get on a hypothetical perfect connection with no error correction. So error correction actually gives you a slight **increase** in speed.

Error correction adaptively alters its internal packet size, depending on the error history. Alas, it can happen that the modem has been excessively optimistic in its choice of carrier speed, resulting in a high error rate on the packets being sent between the modems. The result of that is that the error correcting protocol has to do a lot of retransmission, which reduces the throughput seen by the computer. The computer still does not see any erroneous data, but speed is poor. In the worst case, throughput is a fraction of what it should be for the particular carrier speed. The answer is to force the modem down to a lower carrier speed (using modem commands provided for that purpose). This gives a lower error rate on the packets, less retransmission happens and throughput improves.

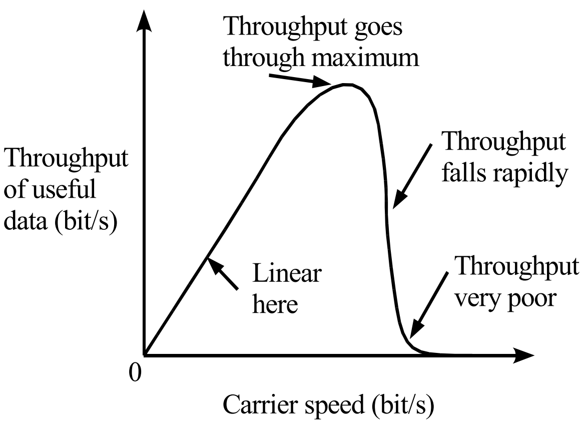

Figure 1

The graph above (Figure 1) is an approximate representation of what happens to the error corrected data throughput as carrier speed is increased. At low carrier speeds the modem has no trouble extracting data, so the error rate is very low, no retransmissions of packets are necessary, and so the throughput increases linearly with increasing carrier speed. Ultimately the error rate increases so the throughput no longer increases and it goes through a maximum. This maximum is the absolute greatest possible throughput possible for the particular telephone connection. Any attempt to further increase the carrier speed results in more errors, more retransmissions and a lower throughput.

The ideal carrier speed is the point where the throughput is at a maximum. Modem buyers are inclined to feel aggrieved if their modem does not connect at the highest possible carrier speed, so chipset manufacturers tend to be a little optimistic in their choice of carrier speed. The graph above shows that this is not really a good idea. It is often useful for the modem user to force a lower carrier speed and thus actually improve throughput.

Data Compression

Quite often, the data through a modem shows repetitive patterns. There is an opportunity for a data compression algorithm to increase throughput. There are two algorithms commonly implemented in modems MNP 5 and V.42bis. V.42bis is the latest and best. The algorithms are completely transparent. The computer does not have to know anything about data compression. All the computer sees is a higher throughput.

Data compression is very vulnerable to errors in the compressed data stream. Data compression must always have error correction running first. A modem will not negotiate data compression if it has failed to negotiate error correction. On connection, a modem first negotiates a carrier speed, then error correction, then finally data compression.

All data compression algorithms work by squeezing out repetition. If the data compression algorithm sees compressible data then the throughput can be higher than what it would be in the absence of compression, with the carrier speed remaining the same. V.42bis claims a maximum compression ratio of 4. In practice, you will very rarely see that level of compression; 1.7 is common on a plain ASCII text file. It is possible to create a test file where the compression is higher than 4, but real files almost never get that high.

Files which have already been compressed by software (for example, ZIP files) have already had all the repetition squeezed out of them, so V.42bis cannot do anything more. Many files on the Internet and BBSs are already compressed. The only improvement in throughput you see is coming from the error correction, not the data compression. Because of that, a few Internet Service Providers turn off data compression in the modem. As a result, the port speed only needs to be a little faster than the highest carrier speed. For a 33600 speed modem, 38.4kbit/s would be satisfactory.

The port speed also can affect throughput. Remember that usually there are 10 bits sent through the port for every byte of useful information. If the port speed is set at, say

19,200bit/s, throughput cannot exceed 1920 bytes per second. It can go lower, simply by something sending slower, for any reason.

For Example: You have a 33600 speed modem, which you have set to connect at a carrier speed of 31200 bit/s, an excellent telephone line and you are transferring a plain text file. What port speed should you pick?

Answer: Your modem should negotiate a 31.2kbit/s carrier with a reasonably low error rate on your excellent telephone line. Your error corrected throughput before compression will be around 31200/9 = 3467 bytes per second. Data compression will give you maybe 1.7 times that. So, throughput is 1.7 * 3467 = 5893 bytes per second. You would like a port speed of 10 times the throughput. Port speed at least 10 * 5893 = 58930bit/s. The next higher standard speed is 115.2kbit/s, so that is the port speed you would prefer.

The preceding was an optimistic calculation. It assumed a substantially error free 31.2kbit/s carrier, a compressible file and both computers involved could keep up. Bearing in mind that most files are already compressed by software, a port speed of 38.4kbit/s will always keep up.

My recommendation to most 33600 speed or 28800 speed modem users is to start with 57.6kbit/s port speed and try it. Get familiar with your software, local service providers, what download speeds you usually get and generally develop confidence. Then, cautiously try 115.2kbit/s; if things improve, stay with it. If you suddenly get lots of retries on downloads or other things go wrong — which indicate that overrun errors are occurring — go back to 57.6kbit/s.

For 56k modems, start with 115200 and try to stick with it. If you have port speed related problems, then going down to 57600 would not be too bad. For 14400 speed modem users, start with 38.4kbit/s, then try 57.6kbit/s.

There are expensive com ports and modems which can support port speeds higher than 115.2kbit/s. That is hardly necessary when the practically achievable carrier speed is presently limited to around 50kbit/s.

Signal Quality and Line Level

A modem can easily get blamed for a problem which is actually caused by the telephone line. To help you resolve where problems are, modems have the %Q, %L and &V1 commands.

The %Q command measures Signal Quality. It is a measure of how difficult the modem is finding it to extract data from the analog signal being received over the telephone line. A result of 0 means no trouble at all, data recovery is perfect, signal quality is excellent. signal quality of 0 is too good to be true at higher carrier speeds. A result of 100 means the modem cannot recover data at all, signal quality is terrible. Depending on the setting of the %E command, the modem is about to hang up, retrain or train down in carrier speed.

The %L command measures Line Level. It reports the incoming analog signal magnitude in minus decibels referred to 0.775V (-dBm). The voltage is from 1 milliwatt into 600 Ohms. Decibels is a logarithmic scale of loudness. An exceptionally loud modem signal might be -10dBm. A very soft modem signal might be -40dBm.

For example %L reports a value of 25, this means that the incoming signal is at a level of -25dBm. Modems typically transmit at -14dBm, the network loss is nominally 6dB, so the line level could be -20dBm, %L would report 20. The network loss is often larger than 6dB, so %L results usually go from 20 (= -20dBm) to 35 (= -35dBm).

The &V1 command makes the modem report many parameters, including TX, RX, SQ and LL. It is a very useful command. For example, here is the result of an &V1 command on a Maestro Woomera modem:

```
TERMINATION REASON..... NONE
LAST TX rate........... 28800 BPS
HIGHEST TX rate........ 28800 BPS
LAST RX rate........... 50667 BPS
HIGHEST RX rate........ 50667 BPS
PROTOCOL............... LAPM
COMPRESSION............ V42Bis
Line QUALITY........... 032
Rx LEVEL............... 015
Highest Rx State....... 67
Highest TX State....... 67
EQM Sum................ 00A3
RBS Pattern............ 00
Rate Drop.............. 00
Digital Loss........... 2000
Local Rtrn Count....... 00
Remote Rtrn Count...... 00
V90 9481834347E5
```

TX is shown after “LAST TX rate”. RX after “LAST RX rate”. SQ after “Line QUALITY”. LL after “Rx LEVEL”. The last line is showing that V.90 modulation is established. The &V1 command will even work after hanging up and will show you the parameters just before the hang up.

A modem has more trouble extracting data from a signal which is corrupted by noise than from a signal relatively free of noise. What matters is the signal to noise ratio. Noise on a telephone line tends to be something imposed by external influences. The softer the signal gets, the worse is the signal to noise ratio.

It is only possible to measure signal quality and line level when there is an incoming signal to measure. That is, the modem must be connected to another modem at the time. In order to issue the %Q and %L commands, it is necessary to be in “on-line command mode”. See your modem manual for details.

Signal quality is the important measure. Line level will really only tell you why signal quality is bad if the line level is exceptionally soft. Signal quality gets worse for a long local loop. The higher the carrier speed the worse the signal quality gets. Forcing a lower carrier speed will improve signal quality.

It is wise to be aware of the approximate signal quality and line level figures that you are expecting when you call the various BBSs or service providers that you normally use. That way, if something changes (usually for the worse) you can tell whether it is something at the far end or something at your end.

Com ports

A com port is an electrical interface which transfers data by one bit at a time. Contrast this with a parallel port which transfers data by multiple bits at a time. Com (communications) ports are traditionally used with modems. A com port is not necessarily slower than a parallel port, due to the difficulty of making a parallel port drive a long cable without various electrical problems. Parallel ports are only useful over short distances.

Com ports are implemented using a Universal Asynchronous Receiver Transmitter (UART) integrated circuit (IC). Most new computers are supplied with two com ports, hence they have two UARTs. In the past, the UARTs were separate 40-pin chips, but are now commonly incorporated as just one part of a multi-function IC on the motherboard. This is done for cost saving reasons.

Also available are com port cards, which typically have the UARTs in the form of traditional 40-pin socketed chips. This is very convenient when troubleshooting or when you may want to change the UART type.

There exist many different types of UARTs, as designed by the semiconductor manufacturers. However, there are only two types commonly found in computers, the 8250 and the 16550AN. They were originally designed by National Semiconductor. The common names given above are abbreviations of the full part numbers. Other manufacturers now make the same UARTs, under their own type numbers.

Windows 9X will report the type of UART in Control Panel, Modems. Click the Diagnostics tab, select the com port which the modem is plugged into, then click More Info.

When the UART is reported as an 8250, that usually actually means it is a 16450. The 16450 chip is pin-for-pin and functionally the same as the 8250 chip which was used on the old IBM PC and XT. The 16450 has shorter I/O read and write times, which helps expansion card designers. The 16450 and 8250 are now rare and obsolete.

One characteristic of the 8250 is that it generates one interrupt to the CPU for every character passed through the port. As the speed through the port increases, then the time available for the CPU to service the interrupt gets smaller and smaller. Ultimately the CPU starts missing interrupts, thereby causing *overrun errors*.

The 16550 was developed to deal with this problem. It has internal 16-byte First-In-First-Out buffers (FIFOs) on both the transmit and receive sides. But, it powers on emulating an 8250, the FIFOs must be turned on by software. Provided the software enables these FIFOs, the CPU can transfer multiple characters on each interrupt and has multiple character times in which to service the interrupt. This greatly reduces the problem of overrun errors caused by missed interrupts. Alas, it does not completely prevent missed interrupts because CPU intervention is still required for flow control, thus it would still be possible for a very long delay in servicing an interrupt to cause a FIFO overflow.

There is a rare 16650 chip which has 32-byte FIFOs and facilities for hardware flow control without needing any CPU intervention at all. Provided it is correctly initialised and driven, this chip completely eliminates the missed interrupt problem.

The four UARTs, the 8250, the 16450, the 16550 and the 16650, are all 40-pin chips which are as pin compatible as possible. It is normally possible to pop a lower specification UART out of its socket and put in a higher one. The 16650 has little software support so far.

Internal modems normally do not have a separate UART chip, the equivalent of the UART is built into the modem chipset. To the software, an internal modem appears to be a UART chip (usually a 16550) permanently connected to a modem. One delightful feature of recent internal modems is that there is an internal link between the receive FIFO and the rest of the modem, such that the receive FIFO will not be loaded with another character if it is already full. That totally prevents overrun errors.

CPU Speed

A faster CPU will generally be able to cope with faster port speeds before missing interrupts. More complex software (for example, graphical interfaces) give the CPU more to do and may have more complex (possibly slower) interrupt servicing routines, thus interrupts may again be missed. The modem can easily get blamed for problems which are actually in the computer.

A 16550 port is usually recommended for a 28800 bit/s or faster modem. But the real problem is missed interrupts. Any method of avoiding missed interrupts will do.

Software

There is quite a lot of poor quality software which users are struggling to use. Many problems blamed on modems are actually due to the software. There are two major problem areas: (1) software not sending the correct commands to the modem, (2) losing data due to missed interrupts.

Bad Device Drivers

One particularly insidious problem can be caused by the writers of device driver software. Another device, such as the hard disk or refreshing the screen, may make interrupts from a com port be missed. This is usually caused by badly written software which has failed to allow for interrupts while executing itself. Sometimes the only way to get round this problem is to stop port traffic while the interfering device is being serviced.

What happens is this:

1. There is an interrupt from the device, say it wants its buffer cleared. The interrupt controller on the motherboard interrupts the CPU. The CPU does an interrupt acknowledge machine cycle and gets the processor interrupt number from the interrupt controller. The CPU also automatically turns off its Interrupt Flag (IF), thus ignoring any further interrupts. The CPU begins executing the device interrupt service routine.
2. The CPU has now entered a critical region. It cannot answer any more interrupts. It must get back to a state where it can re-enable interrupts as soon as possible. It

must either, (a) very quickly service the present interrupt (100 microseconds or less), or, (b) disable the present interrupt only. Then it is safe to turn on the IF again. Many interrupts can be serviced very quickly, so strategy (a) is fine.

3. Alas, some driver writers try to do too much with IF off. They should be using strategy (b), but they do not. What happens then is that a character has come in to a com port and has caused an interrupt, but that interrupt has been ignored for so long that another character has come in and overwritten the first character. This is known as an “overrun error”, caused by the missed interrupt.
4. The CPU turns on IF again. This ends the critical region and interrupts can again be answered. The driver writer should consider the possibility that the driver loses the CPU as soon as IF goes on again.
5. The driver completes whatever processing it needs to do to service its own interrupt. This does not adversely affect any other device because IF is on.
6. The driver re-enables its own interrupt, issues an End Of Interrupt instruction to the interrupt controller and returns from the interrupt.
7. The CPU continues with whatever it was doing before the interrupt from the device occurred.

One possible solution to the poorly written driver problem is to stop the flow of data from the modem (using flow control) before attempting any hard disk traffic or screen refresh, then use the faulty driver, then restart the flow of data from the modem. One way of doing this (with RTS/CTS hardware flow control) is:

1. Turn off RTS, wait for all received data to stop.
2. Perform the function needing the faulty driver.
3. Turn on RTS. Flow of received data will resume.

V.34 and V.FC

There are many “recommendations” put out by the ITU (International Telecommunication Union, formerly known as CCITT, the Consultative Committee International for Telephones and Telegraphs). They all have the form of a letter, a dot and then a number. They are also known as “ITU standards”. All ITU standards may be used by any manufacturer.

V.34 is an ITU standard for modem modulation up to a carrier speed of 33600 bit/s. V.FC is an older proprietary standard for carrier speeds up to 28800 bit/s, from Conexant (which was formerly the Semiconductor Systems division of Rockwell). Conexant is a large American semiconductor manufacturer. Before V.34 was finalised, there were a large number of V.FC modems made. These modems support V.FC and the lower modulations, such as V.32bis, V.32, etc., but they do not support V.34. When a V.FC modem connects to another modem which also supports V.FC, then the two modems will normally connect using V.FC at a carrier speed of 28800 bit/s, assuming a good quality telephone connection.

V.34 is a small further development of V.FC. The modulation technique is almost the same. A new method for the modems to negotiate mutually acceptable speeds, called V.8, has been introduced. A new +MS command is used to control permissible carrier speeds. Many V.34 modems also supported V.FC, but some did not. However, V.FC and V.34 are different. V.34 is not a superset of V.FC. A modem supporting V.34 but not V.FC, cannot connect using either V.34 or V.FC, to a V.FC modem (without V.34). The two modems will connect with the highest mutually compatible modulation, V.32bis or a carrier speed of 14400 bit/s.

It can happen that an ISP changes their modems, such that their old modems supported V.FC, but their new modems do not. That can even happen with a firmware change in the modems at the ISP. Clients with V.FC modems were happy with the old modems, since they were getting a carrier speed of 28800 bit/s. But the new modems force the V.FC modems down to only 14400 bit/s, so the clients are unhappy, due to performance having suddenly got worse. Naturally, the technical support staff at the ISP should be well aware of whether they are still supporting V.FC or not, on what dial-up numbers. Alas, a few of them are ignorant and are unable to correctly advise their clients of the true reason for the reduction in carrier speed. The client’s modem may then be blamed as having developed a fault, which is not the case.

Possible solutions include: (1) Allow the client to connect using a different dial-up number which does support V.FC. (2) Upgrade the client modem to support V.34. That may not be sensible, due to the high cost of the upgrade. (3) Buy a new modem. That is the most common solution, because most owners of V.FC modems are keen to move to V.90 (56k modulation).

V.90 and K56flex

A similar thing to what happened with V.FC and V.34, occurred with K56flex and V.90. The proprietary standard came out first, then the ITU standard came out and took over. K56flex is an earlier proprietary 56k modulation, invented by Conexant and Lucent. The receive carrier speed may range up to 56000 bit/s. The transmit carrier is similar to V.34 modulation and may range up to 28800 bit/s. V.90 is a similar ITU standard, except that the transmit carrier may go up to 31200 bit/s. A large number of K56flex modems were made before V.90 was finalised. Most K56flex modems can be upgraded to V.90. Both K56flex and V.90 are based on the same physics, so their performance is similar.

Due to firmware space limitations, earlier V.90 modems could not also support K56flex. Some other modem features were also deleted, to save space. Later 56k modems have more firmware space, so they can have “dual” capability, supporting both K56flex and V.90. V.90 and K56flex are different, neither is a subset of the other. For example, a K56flex-only analog modem cannot establish a 56k modulation with a V.90-only digital modem. There is no common 56k modulation, so they will be forced down to V.34 at a maximum of 33.6 kbit/s.

Initially, large numbers of K56flex-only analog modems were made. Then V.90 was determined, then large numbers of V.90-only modems were made. Dual modems came after that. Digital modems tend to have the latest firmware since they are owned by ISPs. Most digital modems support both K56flex and V.90, but the firmware for K56flex is not getting as much attention as it used to, so K56flex is becoming less reliable. There have also been changes in V.90 itself, so an analog modem with early V.90 firmware can have trouble with a digital modem with later V.90 firmware.

Analog and Digital Modems

All modems are digital internally, but there are two fundamentally different modem types, analog and digital, depending on the interface to the telephone line. An analog modem is a modem made to work with a normal analog exchange line. An ordinary telephone can also be plugged into an analog exchange line. The signal is carried in the form of a small analog voltage on the two wires which make up the line. Analog telephone lines are subject to a large number of possible impairments, such as poor frequency response, noise, cross talk, and distortion. The whole development of analog modem modulation standards has been devoted to getting higher carrier speeds out of a plain old analog telephone line.

A digital modem is connected to the telephone system by a digital link. They are also known as “Central Site Modems” (CSMs). The digital link carries the 64000 bit/s full duplex data stream which would have resulted if a perfect analog modem had been connected to a perfect telephone line then to a perfect exchange line card. It is all done mathematically inside the Digital Signal Processor which is part of the digital modem. The digital link is usually an ISDN telephone line, but it could be some other digital carrier, such as optical fibre. The function of the digital modem is to convert from computer data on the com port to 64 kbit/s for the telephone system, implementing the modulation, error correction and data compression, the same as in an analog modem. So a digital modem can connect to an analog modem, via the telephone system. To the analog modem, the digital modem appears to be a very good quality analog modem. A digital modem can also connect to another digital modem using no modulation at all, then the full 64 kbit/s full duplex capacity becomes available.

The majority of digital modems are at ISPs, in routers, such as the Cisco AS5300. The router is a very boring little box, sitting in a rack, apparently not doing much. However, it has replaced dozens of “rack modems” with their hundreds of blinking lights. The information formerly communicated by those lights is now done by interrogating the router remotely.

On one side of the router is a LAN cable going off to the rest of the ISP. On the other side is up to 120 ISDN phone lines. Those 120 phone lines go out through 8 pairs of wires to the local telephone exchange. Each pair is carrying a special ISDN high speed modulation, which is carrying the data for many phone lines. At the exchange, all the 64 kbit/s channels are recovered and go off to the various clients.

On the computer side of the digital modems, there is no separate com port, as would exist with an external modem. The digital modems are all internal modems wired to a computer bus inside the router. The LAN interface also runs off this computer bus, so the data can get from the LAN to the modems. Numerous digital modems are made on one printed circuit board, which is proprietary to the router manufacturer.

For a 56k modulation to start, there must be a digital modem at one end (typically an ISP) and an analog modem at the other end (typically an ISP client). There must be a perfect all-digital link from the digital modem all the way to the “line card” in the exchange switch for the client’s analog telephone line. The digital modem effectively has direct control over the Digital-to-Analog (D-to-A) converter in the client’s line card. The analog modem has to measure the sample voltage generated by the D-to-A converter, 8000 times per second. If the analog modem can distinguish 128 different voltages, then 7 bits can be extracted from each sample. 7 * 8000 = 56000 bit/s, so the receive carrier speed would be 56 kbit/s. That is an incredibly difficult feat, because the modem is simultaneously transmitting a V.34-like modulation, plus the telephone line has impairments, plus the line card has a filter which tends to smear one sample into the next. To successfully extract data, the modem has to do an almost perfect job of cancelling out its own transmit voltage, cancelling echoes, equalising frequency response and so on. That takes serious computation in the DSP.

If only 64 voltages can be distinguished. That gives 6 bits per sample. The receive carrier speed would be 6 * 8000 = 48000 bit/s. Similar logic applies for all other receive carrier speeds defined in the modulation.

56k modulation is asymmetric. In the whole path from the digital modem to the analog modem, there must be one only analog link, the client’s telephone line. In the receive direction, there must be only one D-to-A conversion in the telephone system. The modem does an A-to-D conversion, but it is at 16-bit accuracy, so that conversion contributes negligible error. The impairments in the receive direction can be regarded as mostly deterministic, so a high carrier speed can be achieved.

In the transmit direction, there is a rather poor quality 8-bit A-to-D converter in the exchange. Also the exchange does a relatively poor job of separating the transmit voltage from the receive voltage. So it is much more difficult to get a clean signal, so only a relatively lower carrier speed can be achieved. In practice, 56k modulations are biased in favour of the receive speed. That is where the bottleneck is for web browsing, which is by far the most popular Internet application.

An analog modem can receive at speeds up to 56 kbit/s, but does not have the capability to transmit that fast because it does not have a digital connection to the telephone system. For an analog modem to analog modem connection, there are two analog links involved, so no 56k modulation can start. So the best that two analog modems can do is V.34 at up to 33.6 kbit/s.

What is Inside a Modem

Modems have three major ICs, the data pump, the controller, and the firmware. There are also a variety of smaller ICs and other components.

The data pump handles modulation, demodulation and line equalisation. It implements the complicated rules defined in modem modulation standards. The incoming analog signal is digitised by being sampled and converted into 16-bit binary numbers which represent the instantaneous voltage of each sample. Functions like filtering and demodulation are done by digital computations on the measured values.

In the data pump is a powerful Digital Signal Processor (DSP), which is a specialised CPU design optimised for repetitive numeric calculations. It is considerably faster at that task than most personal computers. When demodulating, the modem needs the DSP to cope with analog telephone line deficiencies, such as degraded frequency response, non-uniform group delay, impulse noise, interference, echoes (local and remote), and crosstalk.

Modulation is done by by using the DSP to compute the required sample values to produce the correct transmit voltage, then passing those values on to a D-to-A converter. Modulation is trivial, compared to the difficulties encountered in demodulation.

The controller is a much less powerful general purpose CPU. The controller does things like the error correction, data compression, interpreting the modem's command set and telling the DSP what to do. The controller runs its own computer program which is stored in the firmware.

The firmware is made in the form of a Read Only Memory (ROM) chip, which is part of the modem. In the firmware is controller code and DSP code. The modem manufacturer customises the firmware from the chipset manufacturer. The chipset manufacturer normally allows the modem manufacturer some control over the controller code, but no control over the DSP code. On more deluxe modems, the firmware may be an EEPROM (Electrically Erasable Programmable Read Only Memory, known as "flash memory") which allows the firmware to be changed by downloading new firmware code into the modem. That is known as a "flash upgrade".

Exchange Lines

The performance of a modem depends on the quality and reliability of the telephone connection. The Public Switched Telephone Network (PSTN) is the transmission medium for most modem traffic. In most places, there is an analog link between the subscriber's premises to the local telephone exchange, then there is a digital link between exchanges, then, typically, a digital link to the other subscriber, such as an Internet Service Provider. The analog link consists of two wires and is known as the "local loop". The wires are in "pairs" in telephone cables.

The local loop is the major source of signal degradation on most modem connections. A small vulnerable analog signal travels a distance of typically several kilometres, picking up noise and distortion on the way. The local loop passes through a variety of cables, many connections and near a variety of possible sources of interference on its way from the subscriber to the telephone exchange. Every local loop has its own particular characteristics. Levels of interference, line attenuation and other parameters can vary with the time of day, humidity, temperature and the random activities of other equipment users.

At the exchange the analog signal is digitised into a digital data stream of 64kbit/s, full duplex. "Digitised" means converted from analog to digital for sending out over the network. "Full duplex" means data flows in both directions at the same speed simultaneously. The analog signal is sampled 8,000 times per second to an accuracy of 8 bits per sample. Another digital stream from the other end is converted back to analog, which is sent down the local loop. The "line card" in the exchange does the digitisation. The local loop is carrying two analog signals simultaneously, the transmit signal and the receive signal.

Full size exchanges run by a telephone company use a method of digitisation called Pulse Code Modulation (PCM), which is in accordance with International Telecommunication Union (ITU) standards. Exchange equipment is fully professional, and is tested for conformity to the standards. Modems are designed to deal with the deficiencies introduced by digitisation in accordance with the standards. The best way to connect a modem to the telephone system is to give it its own telephone line all to itself, with no other devices connected to the same telephone line. Things like lightning arresters and extension leads will not adversely affect modem signals, so they can stay connected. Disconnect all other telephonic devices from the line. Making a modem signal go through a PABX is bad, see the later section about PABXs. A line sharer might be bad for modem signals, see the later section about telephone line sharing.

Alas, fallible human beings set up and maintain an exchange and its local loops. You will normally get full modem performance through an exchange, but even a good recent exchange can have a fault which adversely affects modems.

The digital link between exchanges is usually perfect. Sometimes it can happen that there is a shortage of capacity in a locality, so the telephone company may have to use a "pair gain" scheme to push more calls through a limited cable. One such scheme is "Adaptive Pulse Code Modulation" which allows a voice conversation to only use 32kbit/s, full duplex. It is difficult for most people to tell that this has happened to their call. However, modem signals are adversely affected, and the modem will be forced down to a much lower carrier speed. The telephone company may be willing to provide a higher quality connection, possibly at some extra cost (for example, the "Faxstream" service).

Some country areas have old exchanges. International calls are particularly variable. Many countries have a poor telephone network.

Complex Impedance

In 1993 the Australian telecommunications regulatory body ordered that subscriber equipment should present a complex impedance to the local loop, instead of the former purely resistive 600 ohm impedance. The complex impedance is called "TN12" and consists of 220 ohms in series with a parallel combination of 820 ohms and 120 nanofarads. All subscriber equipment should do the equivalent of producing a transmit voltage from an ideal floating voltage source in series with TN12. The receive voltage is taken by an ideal voltmeter across the TN12, with half the transmit voltage subtracted.

If the telephone line was an ideal TN12 transmission line, the result of this technique would be the complete separation of transmit voltage from receive voltage, which is desirable for high speed modem communication. It happens that the average subscriber local loop is closer to a TN12 transmission line than a 600 ohm transmission line. The 600 ohms impedance is a hangover from the old days of open wire lines.

Over the years, telephone exchanges have mostly been replaced or modified to give TN12 on all local loops. There are only a few older exchanges still in service which present a 600 ohm impedance.

Well designed modems present the TN12 impedance to the telephone line.

PABXs

Private Automatic Branch Exchanges (PABXs) are often digital internally, but they normally use a very cheap method of digitisation, such as:

- Continuously Variable Slope Delta modulation (CVSD).
- Adaptive Pulse Code Modulation (ADPCM) at 32kbit/s.

A standard office PABX is unlikely to be satisfactory for modem communication purposes. A high speed modem will not normally deliver full performance through a PABX. Some PABXs have a digital connection between the PABX and its associated extension telephones. A modem relies on being connected to an analog telephone line, it will not work at all on a PABX digital extension line. The digital signals could even damage the modem. Digital PABXs may optionally have installed analog extension lines which are suitable for modems, but performance will be poor.

The best way to deal with PABXs is to avoid them. PABXs have incoming exchange lines. Connect modems and fax machines to exchange lines, preferably directly with no other device on the line. It is possible to share an exchange line between a modem and a PABX, similar to sharing a line between a modem and a fax machine.

Telephone Line Sharing

It is perfectly possible to share a telephone line between a modem and one or more other telephonic devices, in the same way as having two telephones on the one line. On the line side, telephonic devices (such as modems, telephones, etc.) are usually two-terminal devices, polarity insensitive. They are easy to connect in parallel.

A common arrangement is to have a fax machine, a telephone and a modem on the one line, connected in parallel. That way, the device which answers first gets an incoming call. Any device may call out, but a collision could occur if two devices try to call out at the same time. The danger of collisions can be reduced with a manual switch or a line sharer device.

A line sharer (such as the National Communications “Easy-Connect”) automatically switches between, say, an answering machine (with a telephone in parallel) and a modem. There are three modular telephone connectors on the line sharer, one for the exchange line, one for the answering machine, one for the modem. An incoming call at first goes to the answering machine, the answering machine answers and produces its outgoing message. After the answer, the line sharer listens for fax or data calling tone (fax CNG tone, or V.25 data calling tone). If there is no tone, the call proceeds normally and the caller is not aware there is a line sharer at all. If the line sharer does detect a tone, it generates ring voltage for the modem, which answers, the line sharer then switches the call through to the modem. The modem then receives the incoming fax or data call. The result of this technique is that voice, fax and data calls may be received on the one line and automatically switched to the correct device.

There is a more detailed discussion of line sharing, particularly the Easy-Connect Pro, in the document “Easypro.doc”.

A voice modem can do a similar thing to a line sharer with “adaptive answering”. The modem answers an incoming call and can then tell whether the call is voice, fax or data, then handle the different call types appropriately. Voice callers hear an outgoing voice message and may leave a voice message of their own. Faxes are received as normal. Data calls go to a “host mode” in the software, which gives facilities similar to a Bulletin Board Service (BBS). Of course, adaptive answering must be supported by the software.

Conclusion

Modems are a difficult-to-use technology. The future of telecommunications is for consumers to have a direct digital link to their local exchange. There are competing technologies to achieve this, such as ISDN (Integrated Services Digital Network), cable modems (a high speed modem which connects to a pay-TV cable), xDSL (various kinds of Digital Subscriber Line). Conventional modems will then be obsolete, most communications tasks will be easier, many new possibilities will open up.

There is a document, from the same author, called “The Future of Telecommunications” which discusses these matters.

Copyright Notice

Comments on this document are welcomed.