19 Data Communications

It is generally assumed that data communication is a modern aspect of amateur radio, but of course Morse code transmission can also be regarded as a data mode. Recent deregulation of international and local Morse requirements has done nothing to weaken the interest in Morse, and the mode remains one of the most popular on the DX bands.

The mode of operation first associated with data communications by most amateurs is radio teletype (RTTY). Amateurs in the UK first began using RTTY on the air in the late 1950s with surplus machines such as the Creed 7B. Despite inferior performance in comparison to that of more recently developed modes, RTTY still has its enthusiasts. There are still many RTTY contests, although now computers are used in place of mechanical machines.

DATA MODE DEVELOPMENTS

Digital electronics and then computers allowed technically improved data modes to be developed. The first, developed by Peter Martinez, G3PLX, was called AmTOR (from 'amateur teletype over radio'). Based on the commercial SiTOR system, this was the first amateur data communications mode to make use of error-detection and correction techniques, albeit in a very simple form. This was to be followed by many other specially designed data modes. Commercially developed modems offering protocols such as PacTOR, G-TOR and CLOVER were introduced, and these continue to provide error-free (but expensive) HF communications for bulletin-board and automated mail systems; these modes correct errors using two-way transactions. **Fig 19.1** shows how modern modes have developed from RTTY.

There has also been strong interest in modes that (like RTTY) offered the amateur operator real-time keyboard to keyboard chatting. From the mid 1990s, a serious effort was made to replace RTTY with modes designed specifically to take into account the requirements of amateur chat modes, and the characteristics of HF propagation. The first of these new 'designer' modes was PSK31, developed by G3PLX. An amazingly fruitful period of new mode development followed, based on the PC sound card, and as one experienced operator commented:"More modes have been developed in the last few years than in the previous century!" [1]. During this period, Hellschreiber was revived, MT63 was developed, and MFSK16 was introduced. Other less successful modes have come and gone, or remain as curiosities. Some newer developments are used for special applications, for example slow PBSK modes and the MFSK mode JASON (weak signal modes for LF); and WSJT and high speed Hellschreiber (for meteor scatter use).

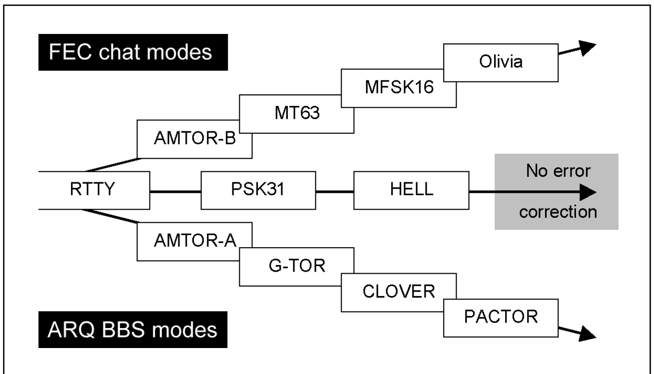

Fig 19.1: Development of HF Digital Modes

The late 1990s and early 2000s saw the ascendancy of the high-speed personal computer with sound card as the preferred method of signal processing. In addition to the development of new modes that would not have been possible without digital signal processing, the sound card technique has made digital mode operation easy and inexpensive for beginners. It has also paved the way for a range of new tools, such as the spectrogram, used to detect weak signals and monitor propagation.

Although not exactly a data mode, Slow Scan Television (SSTV) technology also advanced significantly during this period, to become a popular addition to the SSB QSO. Through the use of the sound card and digital signal processing, inexpensive SSTV operation is now widely enjoyed. See the Image Techniques chapter for more on SSTV.

The trend in HF data modes is continuing to move away from specialised hardware toward general-purpose computers using digital signal processing. Performance never dreamed possible is becoming a reality, and indications are that before long the receiver and transmitter as we know them will be completely absorbed into the computer - the so-called software defined radio [2].

Towards the end of the 1970s, amateurs in North America began experimenting with microprocessor and digital techniques to generate and process data. A standard based on the CCITT X.25 public packet switched network protocol was finally agreed on, and this Amateur X.25 (AX.25) protocol became the standard for amateur packet radio. During the 1980s and 1990s the technique grew into a global amateur integrated network offering error corrected communications between any two stations in a network, file transfer, automatic message storage and forwarding, and bulletin dissemination. While packet radio no longer has the following it once had, largely due to the incredible popularity of the Internet, useful work is still continuing in the area of packet data handling. There is especially strong interest in international message handling for remote users, computer networking, and in technology related to telemetry and the tracking of vehicles and other assets. The leaders in this area have been Bob Bruninga, WB4APR, (APRS) [3]; Phil Karn, KA9Q [4]; Ian Wade , G3NRW (radio network operating systems) [5], and the late Roger Barker, G4IDE (UIVIEW and WINPACK) [6]. Packet radio has also been widely used to disseminate hints to DX operators ("DX spotting").

GETTING STARTED

There is no need to be daunted by the prospect of operating HF data modes. While some systems require skill and experience, there are also some very effective computer modes that are easy to set up, and a pleasure to use. The simplest are Hellschreiber and PSK31.

HF keyboard mode operation is now a very cost-effective addition to the ham shack. Not only is the equipment inexpensive, if an HF rig and computer are available, but also the more effective modes do not require high power or large antennas for good DX. All the keyboard to keyboard modes now use sound card technology, and there is software for the most popular modes

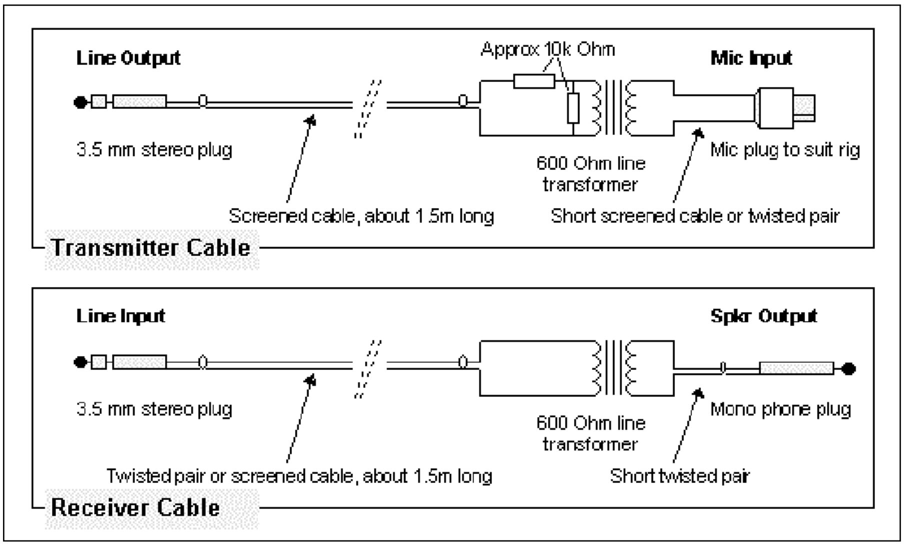

Fig 19.2: The simple cables used to connect PC to radio

available for several different platforms - the most popular being the ubiquitous PC with some version of Windows™ or LINUX™ operating system. Some excellent commercial software provides the widest range of features and operating modes. There is certainly no performance compromise involved in operating using a computer rather than a specialised hardware solution.

To operate these digital modes, a modern computer with sound card is required (computer specification depends on software, but even a 233MHz Pentium will provide a lot of pleasure). Some simple cables, easy to make, are also required, or a commercial interface such as the RIGBLASTER™ can be purchased. A conventional HF SSB transceiver is used, connected via the receiver audio output and microphone audio input. It is best if this is a modern solid-state unit, with good filters and low drift, but many operators use older rigs for RTTY, MT63, Hellschreiber and SSTV with no particular problems. These are the modes least affected by drift and poor frequency netting.

Some of the newer modes require very high stability, and very low frequency offset is necessary between transmit and receive. Most synthesised transceivers will suffice. A transceiver that drifts less than 5Hz per over will operate the newer modes very successfully. Unfortunately offset cannot be accurately corrected by using the Receiver Incremental Tuning (RIT).

The connections between computer and transceiver are quite straightforward. Most amateurs should be able to build the required cables. See Fig 19.2.

The resistors in the transmit cable are used to attenuate the sound card signal so that it does not overload the transceiver. If the microphone socket is used, a lower value of resistor may be necessary across the transformer to further attenuate the audio. If an accessory socket is used, the values shown may suffice.

While the receiver cable is shown with a connector to be directly plugged into the external speaker socket on the receiver, with many rigs this will disconnect the speaker, which isn't helpful. It is best in this case to use an adaptor allowing both the PC cable and an external speaker to be connected.

There is a very good reason for not using the computer speakers instead of an external speaker on the transceiver. Computer speakers receive their audio from the sound card output, and by connecting the LINE IN signal from the radio to the LINE OUT or SPEAKER OUT and the speakers, the receiver output signal will also be sent to the microphone input of the transceiver. This causes feedback problems, especially if VOX is used.

The Importance of Isolation

The transformers shown in Fig 19.2 provide complete DC isolation between the computer and the radio transceiver. The most compelling reason to do this is to prevent serious damage to the radio and computer. Most power supplies are grounded for safety reasons. If the power supply cable to the transmitter becomes loose, the full 20A transmitter current can pass through the microphone circuit, down the cable and through the computer sound card to ground via the PC power cable. Even if the transmitter power cable is considered reliable, significant current could still flow through the sound card cable, causing instability, hum and RF feedback. The simple expedient of isolating the connections also reduces the risk of RF in the computer, and computer noises in the radio.

VOX and PTT Control

Most operators find VOX operation of digital modes quite appropriate and reliable, although the delay may need to be set longer than for Morse or SSB. If for some reason direct control of the rig is necessary, the transmit control must also use an isolated circuit. An opto-coupler does this nicely, driving the Press - to - Talk (PTT) directly without requiring a relay or any further power supplies.

The digital mode software usually controls the transceiver via a serial port, by driving RTS or DTR (often both) positive on transmit, with an appropriate delay before sending tones out from the sound card. The design in **Fig 19.3** is an appropriate PTT circuit for a transceiver with positive voltage on the PTT line and a current when PTT is closed of up to 100mA or less.

Many transceivers include an ‘accessory socket’, offering line-level audio inputs and outputs for transmit and receive. Using these instead of the microphone socket and speaker socket can be really convenient, but can lead to a range of unexpected problems. Sometimes PTT is not available from the accessory socket, and sometimes the VOX does not operate from this socket. The signal levels can also be quite different to the speaker and microphone connections. Even more troublesome, some transceivers leave the microphone operating while sending data through the acces-

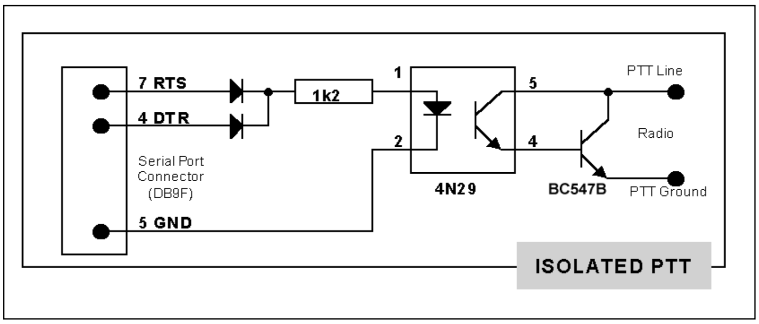

Fig 19.3: An opto-isolated PTT circuit

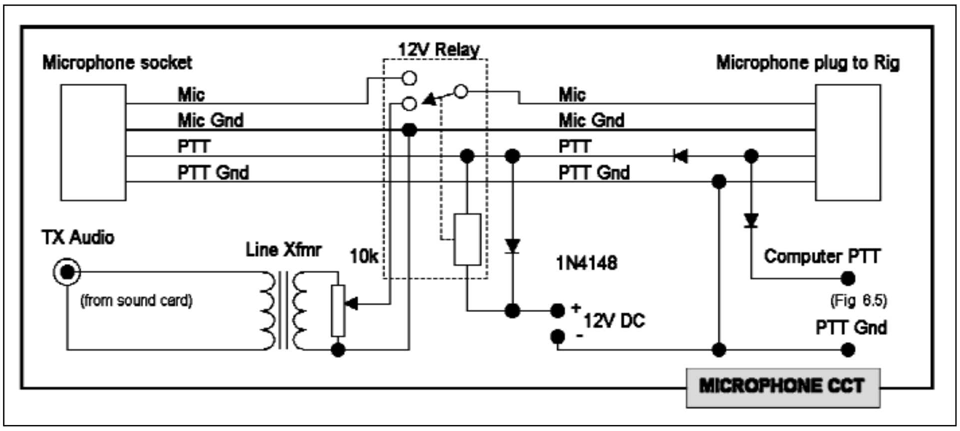

Fig 19.4: A simple way to connect microphone and computer

summary of the specification of the RTTY mode is shown in Table 19.1.

The RTTY technique as we know it took many decades to become what it is today. Engineers Emile Baudot (who developed the system of multiplexing the data), Donald Murray (who developed the actual alphabet), Frederick Creed (who sold telegram printers to the GPO), Howard Krum (who developed the start-stop technique), and finally Edwin Armstrong (who developed the FSK keying technique) all played significant parts. Perversely, the alphabet developed by Murray is also known as the Baudot code, although it wasn't developed by Baudot, and is an alphabet, rather than a code!

sory socket, so coughs, mutterings and keyboard clatter go out over the air!

A simple home-made adaptor (see Fig 19.4) provides a way to operate voice and data modes interchangeably without disconnecting anything. Using this design, the data transmit cable is connected by default, but when the microphone PTT switch is depressed, the relay switches over the audio input and normal microphone use occurs. Operation is simple, and feels natural. The isolated PTT circuit of Fig 19.3 can be built into the same box, and the whole assembly replaces the transmit cable in Fig 19.2. There are several similar designs offered as kits [7].

There are several commercial interface designs available for users not disposed to building a home-made or a kitset interface, but not all provide full isolation. There are also USB interfaces suitable for laptop computers and others with no serial port or sound card.

An area that causes confusion among beginners is the business of setting up and adjusting the sound card. The adjustments are all performed in software, mostly using an application provided with the operating system, and once set for one mode or program, the settings should be correct for all the rest. There are two main software adjustments, for transmit and for receive, and it is not very obvious where to find these, especially the receiver adjustments. The better applications provide direct access to the adjustments. In addition to the gain settings, you need to select the correct inputs and outputs, and disable those not being used. The procedure and these adjustments are described in detail in the RSGB publication *Digital Modes for All Occasions* [8], a reference work recommended for both novice and experienced operators.

RTTY

RTTY is now almost exclusively operated using computers, but the actual data signalling remains the same as it was in the 1950s, when mechanical machines were used.

RTTY uses five sequential pulses to represent each of the letters, figures, symbols and machine functions. Start and stop pulses are added to facilitate serial transmission. This code is now recognised as the International Telegraph Alphabet No 2 (ITA2) [9], an international standard with national variations. A

Symbol Rate	45.45 or 50 baud
Typing Speed	60 or 66WPM
Bandwidth	270Hz
ITU-R Description	270HF1B

Table 19.1: RTTY summary

The ITA2 five-pulse alphabet has only two conditions for each of the five data pulses (binary 0 or 1, space or mark), allowing for 32 different combinations. Because it was necessary to provide 26 letters, 10 figures and punctuation marks, the 32 combinations were not enough, and the problem was resolved by using most combinations twice; once in letters (LTRS) case, and again in figures (FIGS) case. Two special characters were assigned, LTRS and FIGS, to indicate which case was in use. The receiving station continued to use the case indicated by the last received case command character until it received a different one. Control functions such as LTRS, FIGS, CR, LF, space and blank were made available in either case. The remaining 26 have different meanings depending on whether the LTRS or FIGS case is selected.

Each mode in this chapter is accompanied by a spectrogram that illustrates the bandwidth properties and appearance of the signal. Fig 19.5 is the first of these. The spectrogram is a three-dimensional record of the radio signal - frequency vertically, time

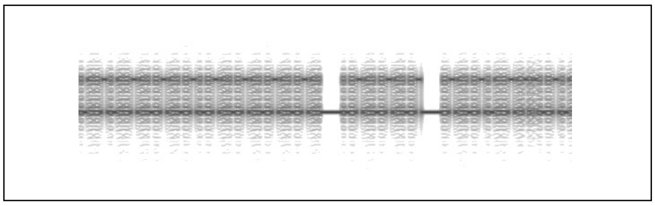

Fig 19.5: The RTTY Spectrogram

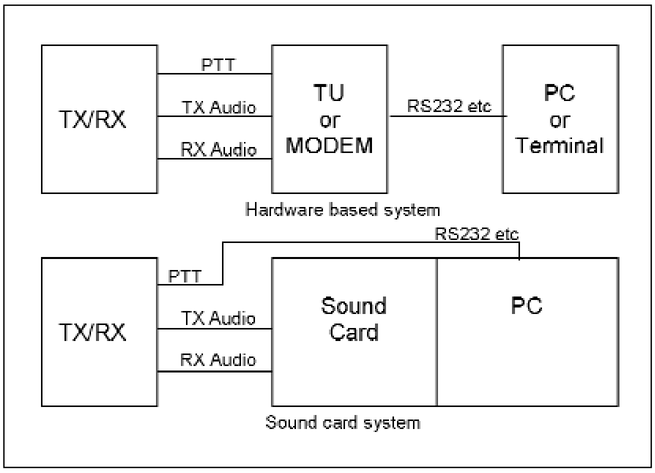

Fig 19.6: Block Diagrams of hardware and sound card systems

horizontally, and signal strength indicated by brightness. All the spectrograms were recorded from live signals at the same bandwidth and time settings, allowing the bandwidth to be assessed and should assist the operator to identify the signal from its appearance.

Compatibility was a problem in the early days of RTTY. Different types of surplus mechanical equipment was being used in different countries, many with different signalling speeds, and the American 45.45 baud speed eventually became the standard. At 45.45 baud each element is 22ms long, and each character takes about 165ms to send, so there are about six characters per second, or 60WPM.

So that the signals can be carried on a radio transmitter, the low frequency data pulses must be modulated onto a carrier or tone (sub-carrier). Similarly, at the receiver, the audio frequency sounds must be demodulated back into pulses. Equipment to do this is called a terminal unit (TU) or modem, although in computer systems a sound card and software inside the computer now perform these functions (**Fig 19.6**). Most operators now use audio frequency shift keying (AFSK) and an SSB transmitter.

AMTOR

RTTY suffers from problems such as multi-path reception (fading) and noise, which make successful decoding difficult or impossible on many occasions. In order to overcome these problems, it is useful to be able to compare multiple versions of the same transmission. This can be achieved by using frequency, polarisation, space or time diversity. AmTOR uses time diversity, sending groups of characters twice, spaced by a small time interval. AmTOR has two modes, Mode A (ARQ - automatic repeat request) and Mode B (FEC - forward error correction), which use this time diversity in different ways. In Mode A, a repeat is only sent when requested by the receiving station, while in Mode B each character is always sent twice.

AmTOR was developed from the commercial SiTOR system, which was devised to improve the communication between teleprinters using the ITA2 alphabet. The system uses a seven-pulse alphabet with an exact correspondence to the ITA2 five-pulse code - the two extra pulses provide error detection information [10]. This Moore code was designed to have a constant ratio of four binary 1s to three binary 0s in all valid combinations, so only 35 out of the possible 128 combinations are valid. This provides a form of error detection since any character which does not have this 4:3 ratio is known to be in error and can be rejected. In addition to accommodating the 32 ITA2 combinations, a further three are available for link information signals (Idle Signal Alpha, Idle Signal Beta and Repeat request (RQ)).

AMTOR Mode A is a synchronous system, which transmits blocks of three characters from the transmitting or information sending station (ISS) to the receiving or information receiving station (IRS). During a QSO, the roles switch as the direction of traffic changes. The ISS sends its message in groups of three characters, pausing between groups for a reply from the IRS. The signalling rate is 100 baud, with each character accounting for 70ms and a three-character block occupying 210ms. The block repetition time is 450ms, so there is 240ms in each cycle when the ISS is not transmitting.

Symbol Rate	100 baud
Typing Speed	66WPM
Bandwidth	400Hz
ITU-R Description	400HF1B

Table 19.2: AmTOR summary

This 240ms period is taken up by the propagation time between stations, time for the IRS to send back its link information, time for the return journey back to the ISS, and an allowance for switching delay from transmit to receive. This time should be less than 20ms. The 450ms block repetition cycle limits the distance over which a Mode A QSO can take place.

A spectrogram of an AmTOR Mode A transmission is shown in **Fig 19.7**, and one showing AmTor Mode B is in **Fig 19.8**. **Table 19.2** shows a summary of the AmTOR specification.

When it is required to transmit to no particular station (for example when calling CQ, operating in a net, or transmitting a bulletin) there is no one station to act as IRS. Similarly, Mode A isn't helpful to others listening in, as they do not get corrections, but do receive repeats they may not need. Mode B is designed for these applications, and achieves a simple forward error correction (FEC) technique by sending each character twice. In order to provide time diversity, each character is repeated after four other characters have been transmitted, thus avoiding errors associated with bursts of noise. The receiving station tests for the constant four to three ratio, and prints only correct characters. If neither version is correct then an error symbol is displayed.

AMTOR Mode A is little used these days, although commercial traffic is still widely heard. Unfortunately timing restrictions make PC sound card programs for AMTOR Mode A impractical. AMTOR Mode B has enjoyed a longer life since it can be successfully operated using a computer with sound card. It is a useful mode for bulletin broadcast.

PSK31

Also developed by G3PLX, and based on an idea by Pawel Jalocha, SP9VRC, PSK31 was intended to replace RTTY as a simple to use and easy to tune keyboard chat mode. The PSK31 mode first used a low-cost DSP starter kit as modem, but is now firmly in the realm of PC sound cards and public-domain software, using modern DSP techniques [11]. The bandwidth of PSK31 is much lower than most other data modes, and has high sensitivity, which means it can work at lower signal levels in today's crowded bands.

Keying is achieved by phase-shifting the carrier by 180°, rather than frequency-shifting it, resulting in a very narrow-band signal. The technique is called differential binary phase shift keying, or BPSK. Data is encoded in the phase difference, rather than absolute phase, since phase is not constant due to ionospheric effects. With the chosen baud-rate of 31.25, the bandwidth is down from the 300-500Hz of other modes to only about

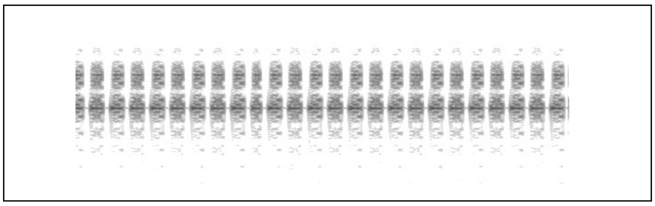

Fig 19.7: The AmTOR-A Spectrogram

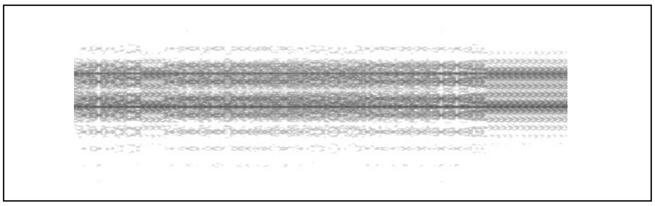

Fig 19.8: The AmTOR-B Spectrogram

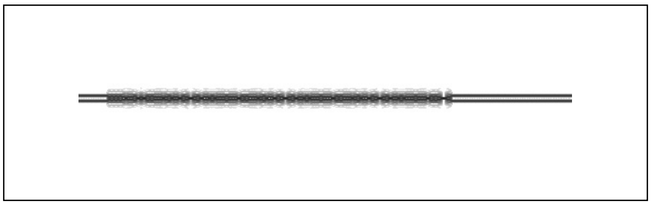

Fig 19.9: The PSK31 Spectrogram

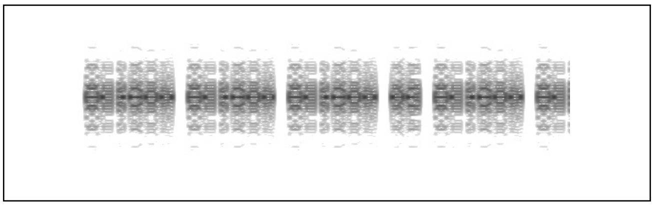

Fig 19.10: The Feld-Hell Spectrogram

Symbol Rate	21.25 baud
Typing Speed	~35WPM
Bandwidth	60Hz
ITU-R Description	60H0J2B

Table 19.3: PSK31 summary

Symbol Rate	112.5 baud
Typing Speed	25WPM
Bandwidth	350Hz
ITU-R Description	350HA1C

Table 19.4: Hellschreiber summary

62.5Hz. By using an alphabet with properties similar to Morse, ie with short codes for common letters, the text speed of PSK31 is about 35WPM. By using narrow filters in the receiver, the performance of PSK31, even without error correction, is certainly better than RTTY and AMTOR. In addition, the transmitted signal is carefully shaped to minimise bandwidth as the phase is switched [12].

A spectrogram of a PSK31 transmission is shown in **Fig 19.9**, and **Table 19.3** gives a summary of the specification.

PSK31 has no error correction, but on average gives much better reception than RTTY. Radio paths with fading and especially high phase shift caused by ionospheric Doppler effects can prove very difficult, as the incidental phase shift can easily exceed the differential phase shift of the intended modulation. A version of PSK31 with convolutional error correction (QPSK31) works well on VHF and paths with burst noise, but is even more adversely affected by Doppler on HF. This is because the phase shift is reduced to 90° in order to accommodate twice as much data (Quadrature PSK). Several other successful variants exist - for example FSK31, developed by UT2UZ, which uses MSK modulation, and PSK63F, developed by IZ8BLY, which uses convolutional FEC on a single 62.5 baud PSK bitstream (rather than QPSK).

PSK31 is popular on the HF bands, and is probably the most widely used digital mode. It is also effective on VHF. The latest software is easy to use, and little transmitter power is required for good DX. Perhaps the most popular spot is around 14.070MHz on the 20m band.

HELLSCHREIBER

There is a grey area (pun intended) between digital data modes (such as RTTY and PSK31) and analogue data modes (such as SSTV or HFFAX), of modes which cannot adequately be described as totally analogue or totally digital.

Arguably, Morse is one of these modes, because while it is apparently a digital transmission, reception occurs at an analogue human-readable level. For example, experienced Morse operators can identify the sender by his ‘fist’, can read the signal better than any electronic means, and can tell much about propagation and the transmitter from the sound of the signal.

Of the other modes in this category, Hellschreiber is the favourite. The term Fuzzy Modes has been coined to describe these modes with both analogue and digital features. Fuzzy modes use the human brain to assist in interpretation of an analogue presentation of the received signal, rather than electronic decisions made by hardware or computer.

While Hellschreiber is a relatively recent arrival on the amateur scene, its origins are old. It was developed as a means of sending press messages by telephone line. Hellschreiber, even in the 1930s, was an audio sub-carrier mode, and was soon sent by radio, predating RTTY for this purpose by at least 15 years. The mode was used to send press traffic right up to the 1960s.

Developed by Rudolf Hell [13], the technique involves sending each character as a pattern of timed dots, rather like a dot matrix printer [14]. Black dots are sent (key down) and white spaces are not sent (key up) by scanning each character vertically upwards, then moving along from left to right at a constant rate. The column data can be analogue, but typically consists of 14 black or white dot positions per column, and seven columns per character, including the space between characters. Each character takes 400ms to send, so the typing speed is about 25WPM. At the receiver, the incoming dots are presented to the reader as dots of varying greyness according to signal strength, allowing the reader to discern the transmitted text by eye, and so enabling text to be recognised in the presence of considerable noise.

A number of clever techniques were developed to improve reception and minimise transmission bandwidth. For example, the received dots are displayed twice, spaced vertically, making synchronisation unnecessary. The font developed by Hell did not permit individual pixels to be sent, rather at least two were sent consecutively, so the characters could have 14 x 7 resolution with the bandwidth of a 7 x 7 font. These features are retained today in PC sound card software for Hellschreiber. Modern techniques include rendered characters (grey pixels on corners), raised cosine dot shaping for minimum bandwidth, and proportional fonts, which are faster to send.

The most popular Hell mode is the original one used over military radio links from 1944 by portable mechanical machines such as the Siemens A2, and for that reason is called Feld-Hell. The signal is on/off keyed at 122.5 baud with carefully shaped dots, and has a bandwidth of about 350Hz. Feld-Hell is especially useful on noisy bands, and because the transmitter duty cycle is only about 20%, is ideally suited to QRP and portable operation. It is badly affected by multi-path, which causes interesting ghosting effects. These can often be minimised by careful adjustment of receiver gain. Another popular mode is FM-Hell, developed by Nino Porcino IZ8BLY, which uses minimum shift keying (MSK), has similar bandwidth, but is more robust and sensitive. FM-Hell is not so affected by multi-path, but operates the transmitter at 100% duty cycle.

A spectrogram of a Feld-Hell signal is shown in **Fig 19.10**, and **Table 19.4** shows a summary of the specification.

There are many free software packages for Hellschreiber modes, the most popular being IZ8BLY Hellschreiber, written by Nino Porcino IZ8BLY. This software includes several other interesting Hell-related modes. Hell signals can be found on most

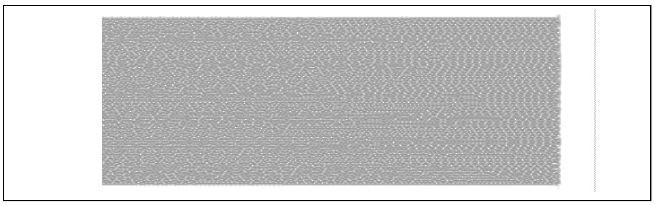

Fig 19.11: The MT63 Spectrogram

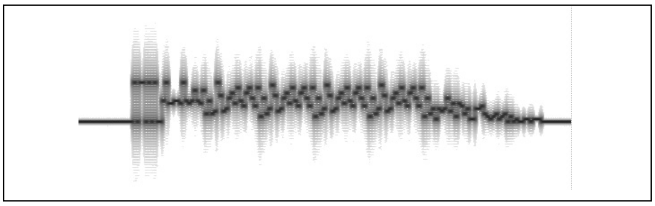

Fig 19.12: The MFSK16 Spectrogram

Symbol Rate	10 baud
Typing Speed	100WPM
Bandwidth	1000Hz
ITU-R Description	1K00J2DEN

Table 19.5: MT63 summary

Symbol Rate	15.625 baud
Typing Speed	40WPM
Bandwidth	316Hz
ITU-R Description	316HF1B

Table 19.6: MSK16 summary

bands, and is most popular on 80m and 20m. Check around 14.075MHz.

MT63

This remarkable mode has been likened to a juggernaut driving down the high street at rush hour - nobody dares get in its way! Developed by Pawel Jalocha, SP9VRC, this is definitely *not* a mode to be used on a crowded band, but it has some very special properties. Not unlike many PSK transmissions at the same time, MT63 uses 64 carriers, spaced 15.625Hz apart, each one phase modulated at 10 baud. The resulting signal bandwidth is 1kHz, and the signal sounds just like noise [15].

What makes the mode particularly unusual is the way the data is coded. The raw data rate is 640 bits per second, but a very strong FEC system is used, and the resulting text rate is 100WPM. The FEC system uses a Walsh-Hadamard transform, where the seven ASCII text data bits index a table of carefully selected 64-bit words to be transmitted. These 64 data bits are then spread across the 64 tones, and also spread over six seconds of transmission. The result is a sensitive mode of incredible robustness, reasonably immune to burst noise and interference, and also able to operate under conditions of ionospheric instability that would stop most other modes.

A spectrogram of MT63 is shown in **Fig 19.11**, and **Table 19.5** shows a summary of the specification.

Because of its bandwidth and slow turnaround (12 seconds between overs!), MT63 is little used for DXing. However, little transmitter power is needed and tolerance to drift and mistuning is about 50Hz. MT63 is a good choice for maintaining regular contact with friends over trans-polar and long path routes. SP9VRC has recently developed an MFSK mode of similar bandwidth which uses the same FEC system. This new mode, named OLIVIA, is one of the most sensitive modes designed yet, and has a typing speed of about 17.5WPM. Both modes have wider and narrower variants, and both can be found regularly just above 14.1MHz.

MFSK16

Multi-frequency shift keying (MFSK) operates like RTTY, but instead of just two tones, four or more are used, allowing more data to be sent at a lower keying rate. MFSK has been used commercially since the 1950s when the Coquelet electro-mechanical system was developed in Belgium. A more sophisticated electronic system called Piccolo was developed for the British Foreign and Commonwealth office and publicly demonstrated in 1963. These two systems were initially designed for the ITA2 alphabet, and converted the signals to and from standard teleprinters. Over the years a number of versions of Piccolo with differing numbers of tones and different speeds were developed, and all exhibited good sensitivity and considerable superiority to teletype on long distance circuits. These systems operated without FEC. It made sense therefore to develop an improved MFSK mode for amateur use.

The benefits of MFSK are:

- Sensitivity improves with the number of tones used.
- More data can be sent as the number of tones increases.
- Since the same data can be sent at a lower keying rate, immunity to multi-path effects is improved.
- Immunity to interference depends on the signalling rate, not the overall signal bandwidth, improving sensitivity and robustness.

These benefits mean that a very robust typing-speed mode can be developed that has high sensitivity. Multi-path reception causes timing errors which cause individual keying elements to run into each other. Because the keying rate of MFSK is much lower for the same data rate, the technique is useful for avoiding the timing problems that cause such difficulty to other modes. One disadvantage of MFSK is that it requires rather accurate tuning and stable equipment.

The first and most successful amateur MFSK mode is MFSK16 [16], designed by Murray Greenman ZL1BPU, specifically for long-path keyboard to keyboard operation. It has also proved to be excellent on 80m where NVIS multi-path problems are extreme. The first MFSK16 computer program was STREAM by Nino Porcino, IZ8BLY, and it is still the most popular. MFSK16 uses 16 tones spaced 15.625Hz apart, and a signalling rate of 15.625 baud. Because each tone represents four bits of data, the data rate is 62.5BPS. MFSK16 uses a very powerful convolutional code FEC system, with an interleaver to provide time diversity. Both the FEC code and interleaver are tied to the data bit weighting, and so no synchronism is required. MFSK16 also uses a variable length character set like Morse and PSK31, which results in a typing speed of over 40WPM.

Fig 19.12 shows the spectrum and **Table 19.6** gives a summary of the specification.

MFSK16 is one of the best DX modes, requiring little power, and capable of operating under rather poor conditions. Users report reception with no errors even when the signal cannot be heard! The FEC system ensures that copy is virtually perfect until printing simply stops as the signal is lost, which cannot be said for most other modes. There are numerous computer programs offering MFSK16. Most operation is close to where RTTY is found, for example just below 14.080MHz on 20m.

The sound of MFSK16 is distinctive, and although tuning the signal is tricky, users soon learn to align the receiver with the lowest (idle) tone. This tone appears at the start of each over and during pauses in the transmission (see **Fig 19.12**).

Other amateur-developed MFSK modes include: MFSK8, also by IZ8BLY; THROB by Lionel Sear, G3PPT; OLIVIA by SP9VRC; FSK441 by Joe Taylor, K1JT; Domino by Con Wassilief, ZL2AFP; and JASON, a narrow band LF mode by Alberto deBene, I2PHD. FSK441 is a high speed four-tone mode for meteor scatter use. JASON and Domino use Incremental Frequency Keying (IFK), encoding the data as differences in frequency, rather than absolute frequency.

MFSK8 is the same bandwidth as MFSK16, but uses 32 tones spaced 8Hz apart at 8 baud. It is extremely difficult to tune accurately. THROB uses an unusual combination of single and dual tones to encode a restricted character set, and operates at 1, 2 or 4 baud. Despite the very low signalling rate, the typing speed is reasonable, since each signal is a complete character. There is no FEC.

Domino is designed for HF band chatting, and encodes each character of a limited (6-bit) character set into two successive tones. The tones are in two interleaved sets of eight, odd and even, and the data is recovered by measuring the distance between successive tones. As you can imagine, if one measurement is in error, the next will be in error in the opposite direction - this is the main flaw with IFK. The receiver synchronises easily because of the odd-even tone sets, and the order of the tone sets is determined by analysis of the received data. Domino has no error correction in its experimental form, and yet it is remarkably robust and forgiving. The ZL2AFP software is very easy to use.

The particular advantage of the IFK technique is much reduced sensitivity to drift and poor tuning. For example, Domino can be received while the receiver is slowly tuned across the signal! Other similar modes with IFK coding and FEC are likely to be developed in the future.

HF ARQ MODES

These modes were developed to provide improved automatic operation on HF. When forwarding mail or in communication with a bulletin-board system, it is important for communications to be letter perfect, or the commands could be misinterpreted or data corrupted. Since these operations are invariably station to station and automated (rather than nets or broadcasts), an ARQ mode is more appropriate.

The first automated systems used AMTOR, which maintained links well, but data rate was poor. Some also used HF (300 baud) packet, which performed very poorly unless propagation was perfect. Most systems now use commercial (and not inexpensive) modems, which operate the specialised modes PACTOR, PACTOR2, PACTOR3, CLOVER II and G-TOR.

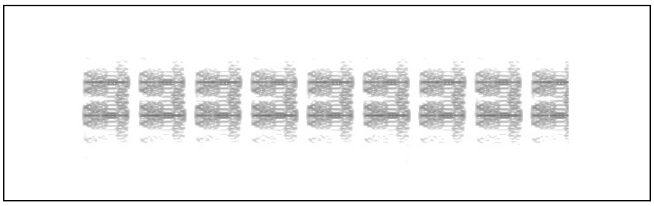

Fig 19.13: The PACTOR Spectrogram

Symbol Rate	100 or 200 baud
Typing Speed	66 wpm (300WPM PACTOR 2)
Bandwidth	500-600Hz
ITU-R Description	600HF1B (500HG1B PACTOR2)

Table 19.7: PACTOR summary

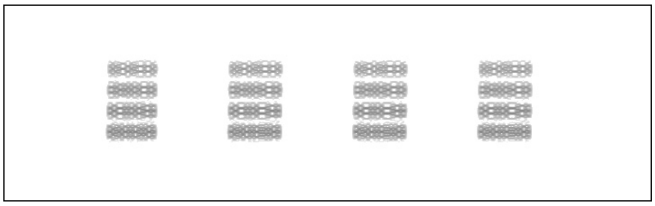

Fig 19.14: The CLOVERII Spectrogram

Symbol Rate	31.25 baud
Typing Speed	30 - 500WPM
Bandwidth	500Hz
ITU-R Description	500HJ2DEN

Table 19.8: CLOVERII summary

The original PACTOR mode is FSK, not unlike AmTOR, except that the data is ASCII, transmitted in longer blocks (1.25s period) and much better error detection is used. In addition, a scheme known as Memory ARQ allows data to be corrected by processing multiple corrupted versions of the same data. Compression techniques are used to reduce the number of bits transmitted.

A summary of the PACTOR specification is in **Table 19.13** and a spectrogram is in Fig 19.13.

Later versions PACTOR2 and PACTOR3 use PSK modulation on multiple carriers, and are considerably faster and more robust. The calling and linking functions retain the original PACTOR FSK modulation mode for compatibility. It is important to appreciate that these are commercial and proprietary modes (not public domain) and therefore their amateur use may be prohibited or restricted in some countries. A special hardware modem is required, and it is also not possible to "listen in" to a transmission in these modes.

CLOVER II has a wide range of different modulation schemes, but is best described as an orthogonal frequency division multiplex (OFDM) system. There are four tone frequencies, each amplitude and phase modulated. Special hardware is required, and the equipment can automatically switch between the available modes in an attempt to provide best throughput. Clover uses Reed-Solomon FEC in addition to its ARQ system. The protocol is proprietary. It is not now used very widely.

Fig 19.14 shows a spectrogram of CLOVERII, and a summary of the specification is in **Table 19.8**.

Of the ARQ modes, G-TOR is the most similar to AMTOR. It has the same FSK modulation, but differs in using the ASCII character set, and in the use of a very strong Golay FEC error correction system, which transmits two differently coded versions of the data. Requests for repeat are reduced because the system is often able to reconstruct the data from the first transmission, and if the second is required the ability to reconstruct the data accurately is enhanced further. G-TOR is proprietary and only available using suitably equipped hardware. Although a good system, unfortunately G-TOR has never enjoyed wide popularity.

PACKET RADIO

Packet radio was the first true amateur digital, as opposed to analogue, transmission system. This makes the relaying of signals much more efficient since the data is reconstituted at each stage of the link and any end-to-end noise and distortion is simply that of the digitising process and not the transmission of the

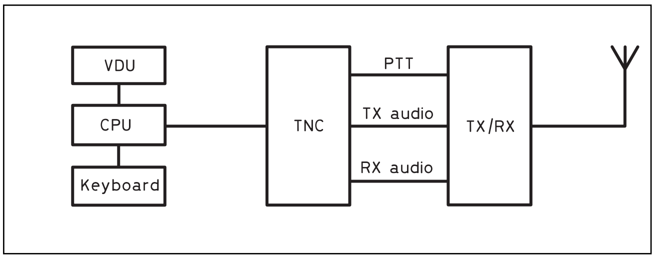

Fig 19.15: Block diagram of a typical packet radio station

digital information. One of the other main benefits of this mode of operation was always assumed to be that the channel could be shared by many users. Unfortunately the radio-based systems are different to computer networks in that not all stations can receive each other, thus making it more difficult for channel sharing.

As with other methods of data communications, packet radio commonly makes use of a terminal unit (Terminal Node Controller, or TNC), either a stand-alone unit or as part of a PC-based system using the sound card as an interface.

Very simply, the function of the TNC is to take the arriving data and assemble it into packets which are then passed to the on-board modem (or PC sound card under PC control) for conversion into audio tones. The receive side of the TNC performs the reverse of the tasks outlined. On VHF the transmission speed for most end-user access is 1200 baud with tone frequencies of 1200Hz (mark) and 2200Hz (space), with 300 baud and 200Hz shift being employed for HF applications. These standards coincide with Bell 202 and 103 modems for VHF and HF respectively. 9600 baud is commonly used for inter-site linking and satellite communications on VHF and UHF.

A block diagram of a typical packet station is shown in **Fig 19.15**. Although the drawing shows a computer, a simple dumb terminal can be used; however, to make use of the full facilities for file transfer etc a computer is essential.

Channel Access

The basis of a packet radio contact is that each station transmits some information and receives an acknowledgement. If no acknowledgement is received then the information is retransmitted. One of the main causes of non-receipt of acknowledgement is collision with another transmission of either the main transmission or the acknowledgement.

Early packet radio experiments made use of a channel access system in which a station transmitted without checking if the channel was free. If the transmission was not acknowledged within the correct time slot, the TNC waited a random length of time before retrying. Current packet systems make use of data carrier detect (DCD) - they listen for an empty channel before transmitting. This is not a guarantee against collisions, because two stations may ‘decide’ to transmit at the same time, but it is an improvement.

AX.25 Level 2 Link Layer Protocol

Version 2 of the AX.25 Level 2 protocol was adopted by the ARRL back in October 1984. This protocol follows that of CCITT Recommendation X.25 except that the address field has been extended to accommodate amateur callsigns, and an Unnumbered Information (UI) frame has been added. This protocol formally specifies the format of a packet radio frame and the action a station must take when it transmits or receives such a frame.

At this link layer, data is sent in blocks called frames. As well as carrying data, each frame carries addressing, error checking and control information. The addressing information carries details of the station which sent the frame, who it is intended for and which station should relay it. This forms the basis of many stations sharing the channel since any station can be set up to monitor all frames on the channel, through various stages to monitor only those intended for it and ignore any others. The error-checking information allows the intended recipient to determine if the frame has been received free of errors. If this is the case and the two stations have previously established a connection, an acknowledgement is generated by the receiving station. If errors are detected the frame is ignored and some time later the sending station resends the frame.

AX.25 Format

Packet radio transmissions are sent in frames with each frame divided into fields. Each frame consists of a start flag, address field, control field, network protocol identifier, information field, frame check sum (FCS), and an end flag. **Fig 19.16** shows the format of a frame and **Fig 19.17** shows a typical address field.

Flag field

Each frame starts and ends with a flag which has a particular bit pattern: 01111110. This pattern appears only at the beginning and end of frames. If five 1 bits show up elsewhere in the frame, a procedure called zero insertion (more commonly called bit stuffing) takes place and a 0 is inserted by the sending station

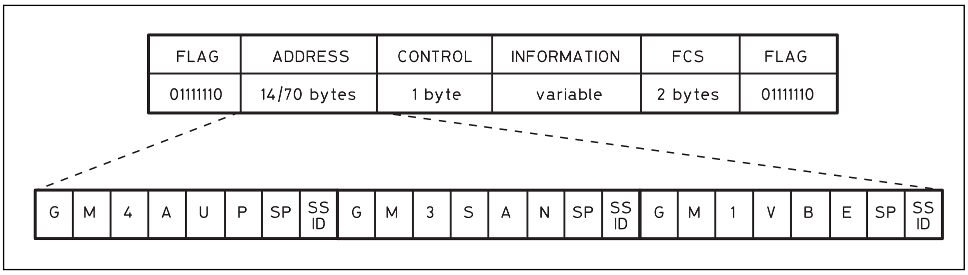

Fig 19.16: Format of a frame

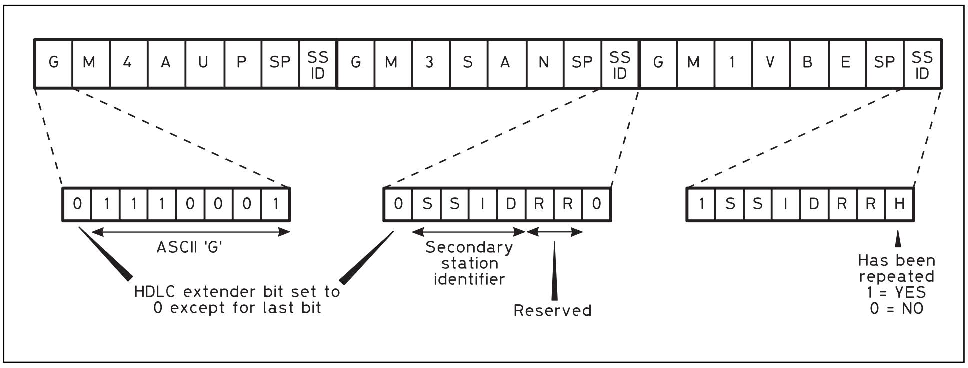

Fig 19.17: A typical address field

and deleted by the receiving station. The receiver will therefore delete any 0 bit which follows five consecutive 1 bits that occur between the flag fields.

Address field

The address field consists of the destination field, source field and up to eight optional relay or digipeat stations. These fields usually contain callsigns and space is available for up to six characters per callsign with a seventh available as a secondary station identifier (SSID). This allows up to 16 different packet radio stations to operate with one callsign. The default is an SSID of 0. For example, GM4AUP-0 could be the real-time station, GM4AUP-2 could be a personal message system (PMS) and GM4AUP-4 could be a node station. The SSID byte in the digipeater address also contains information as to whether it is repeating a frame or not.

Control field

The control field is used to identify the type of frame being transmitted and the frame number.

Protocol identifier field

This field is contained within the information field and identifies what, if any, network layer protocol is being used.

Information field

The information field contains the data to be transmitted and can contain any number of bytes, up to a maximum of 256, of information.

Frame checksum field

The FCS is a 16-bit number calculated by the sender. On receipt of a frame the receiving station calculates a FCS and compares it with that received in the FCS field. If the two match then the receiving station acknowledges the frame.

AX.25 operation

As previously described, the TNC is the device which assembles the data into frames as above. When first powered up, the TNC is in a disconnected state and is monitoring traffic on the appropriate radio channel.

In order to communicate with another station it is necessary to enter the connected state. This is done by issuing a connect frame which contains the callsign it is requesting connect status with as the addressee. If the other station is on the air it responds with an acknowledgement frame and the stations become connected. If no acknowledge frame is received the requesting station re-issues the command a pre-determined time later and continues to do so until a preset number of tries has taken place. If no connection is established the requesting TNC issues a failure notification.

Once a link is established the TNCs enter the connected or information transfer state and exchange information and supervisory frames. The control field contains information about the number of the frame being sent and the number of the last one received (0 to 7). This allows both TNCs to know the current link status and which to repeat if necessary.

When in the connected state either station may request a disconnection which occurs after an acknowledgement is received or if no response is received after several attempts.

Packet Operation

Packet operation currently makes use of the HF, VHF, UHF and SHF parts of the spectrum with both terrestrial and satellite links being utilised. In the early days, much packet operation was real-time person-to-person operation, either direct or through a digipeater.

Most TNCs are capable of digipeat operation and this enables stations who cannot contact each other direct to do so by the on-frequency retransmission of the digipeater. As packet became more popular the real-time operation tended to be replaced terrestrially by store-and-forward systems such as nodes, although there are Earth-orbiting digipeaters placed into operation periodically from amateur-radio equipped space stations.

Digipeaters

Most TNCs can be used as a digipeater as this function is usually contained within the AX.25 Level 2 firmware.

Fig 19.18 shows how two stations A and D can connect to each other using digipeaters B and C. In order for information to be passed from station A to station D via the digipeaters B and C, the information frame must be received by station D and the acknowledgement frame received by station A before a frame can be said to be successfully sent. Digipeaters B and C play no part in the acknowledgement process; they merely retransmit any frames that contain their callsigns in the digipeat portion of the address field. If the acknowledgement is not received by station A then the frame is retried over the whole path. The use of digipeaters has reduced dramatically in recent years with the advent of the network nodes.

Network Nodes

The network node significantly improved the packet radio system as a means of communicating between packet-equipped stations in both real time and by the use of mailboxes. The major advantage of a network node over a digipeater is that any frame which is being transmitted is separately acknowledged between each individual element rather along the whole chain.

Fig 19.19 shows a system with station A trying to communicate with station D via the nodes B and C. In trying to communicate with each other the information is sent from station A to node B and acknowledged back to station A. Node B then passes the frame on to node C and receives an acknowledgement back. Node C then passes the frame to station D who acknowledges it back to Node C. If anywhere in the path no acknowledgement is received then the frame is retried only over the part of the path for which no acknowledgement has been received.

There are two types of network protocol in use - virtual-circuit and datagram. In the virtual-circuit protocol the appearance of a

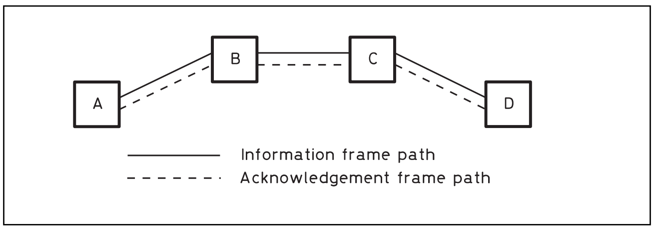

Fig 19.18: How two stations can connect to each other using digipeaters

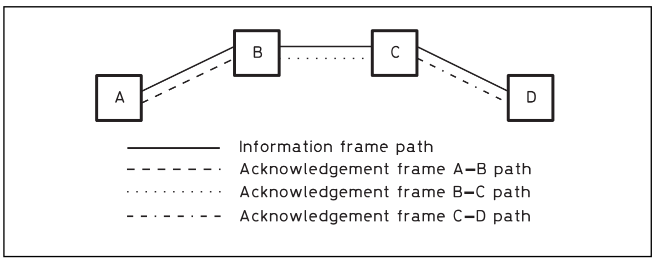

Fig 19.19: How two stations can connect to each other using network nodes

direct connection between the two stations is provided. In order to establish communications a 'call set-up' packet is sent through the network to make a path to the other station. Once this path is established information is sent through the circuit. Any packets sent do not have the full address of the required path because the network attempts to maintain this path for the duration of the contact. After the contact is completed the virtual circuit is cleared by removing the information on the path along the network. An example of a virtual-circuit protocol is the RATS Open System Environment (ROSE) developed by the Radio Amateur Telecommunications Society (RATS) of New Jersey. ROSE is a firmware replacement for TNC2 clones. The virtual-circuit protocol is not very common in the UK and most networking is done using the datagram protocol.

In the datagram protocol each packet contains full network addressing and routing information. This enables a packet to reach its destination via any route still open, regardless of how reliable the network may be. The network overhead is greater in this protocol but it has much greater flexibility and the end user does not need to know the route, only the node nearest him and the node nearest the station with which he desires to connect. Datagram protocols used in the UK are NET/ROM (and clones such as TheNET), TheNODE and Internet.

TCP/IP

The Internet protocol software was written by Phil Karn, KA9Q, and is more commonly known as TCP/IP which is an acronym for two protocols, the Internet Protocol (IP) and the Transmission Control Protocol (TCP).

In reality KA9Q's TCP/IP consists of a suite of individual protocols, Address Resolution Protocol (ARP), File Transfer Protocol (FTP), Serial Line Transfer Protocol (SLIP), Simple Mail Transfer Protocol (SMTP), Telnet Protocol, User Datagram Protocol (UDP) as well as TCP and IP.

Each station using TCP/IP is a network node with a unique IP address that has been assigned by the local IP address co-ordinator. The amateur TCP/IP network has been assigned the network name AMPRNET and all amateur addresses commence with the two digits 44, followed by three digits indicating the country code (as an example of a full address '44.131.5.2' is assigned to G3NRW). TCP/IP is becoming very popular in the UK and is said to offer many advantages over 'ordinary' AX.25.

DX Clusters

A DX Cluster provides information on DX stations being worked/heard along with information on QSL managers, WWV propagation and prefixes for example. The operation is not dissimilar to mailbox operation but users stay connected to their local DX Cluster for as long as they wish to receive announcements. The type of announcement the user receives is customised to suit his own needs and can be used to select prefix information, band information, mode information or a combination of all those.

Each cluster is generally referred to as a cluster node and these nodes can be connected together to each other via the packet network. This enables an item of DX information (commonly referred to as a spot) to propagated to all other cluster nodes in the network, thereby in theory enabling all connected users to see this spot in a short timeframe.

DX Cluster spots are quite common on virtually all contesting software, in lots of major SSB and CW contests, as well as being in use by all major DXpeditions. Some stations remain connected to the Cluster all the time and have an audio warning from their PC to tell them about any DX that might be available. Internet-based DX Clusters are commonly used as an alternative to radio-based DX Cluster connections.

Satellite Communications

Using satellites for communications can be a very satisfying achievement. There are several data satellites orbiting the Earth. Using a dedicated set-up, it is possible to run an automatic station to track, send and receive mail to several different satellites. Full duplex mode is used, and 9600 baud is the standard used. Packet signals have even been bounced off the Moon, although the distortion on that path prevents regular communication.

More information can be found in the chapter on satellite communication.

Packet Radio Bibilography

Further information can found from the following books and periodicals:

Your First Packet Station, Steve Jelly, G0WSJ, RSGB, 1996.

Packet Radio Primer, Dave Coomber, G8UYZ, and Martyn Croft, G8NZU, RSGB, 2nd edition 1995.

NOSIntro, Ian Wade, G3NRW.

AX.25 Link Layer Protocol, ARRL.

ARRL Handbook, ARRL.

RadCom, the RSGB members' magazine.

Information on licensing and policy matters with regard to data communications is available from the chairman of the Data Communications Committee c/o RSGB, Lambda House, Potters Bar, Hertfordshire EN6 3JE.

REFERENCES

[1] Comment made in conversation by Dr Gary Bold ZL1AN

[2] See http://www.sdradio.org/

[3] See http://web.usna.navy.mil/~bruninga/aprs.html

[4] See http://www.ka9q.net/

[5] *NOSIntro: TCP/IP over Packet Radio*, Ian Wade, G3NRW, TAPR

[6] See http://www.ui-view.org/ and http://www.winpack.org.uk/

[7] The BARTG publish designs from time to time. See http://www.bartg.demon.co.uk/

[8] Appendix D, *Digital Modes for All Occasions*, Murray Greenman ZL1BPU, RSGB

[9] A table of the ITA2 character set is to be found in Digital Modes for All Occasions, Murray Greenman ZL1BPU, RSGB

[10] A table of this character set is to be found in Appendix B, Digital Modes for All Occasions, Murray Greenman ZL1BPU, RSGB .

[11] See http://www.aintel.bi.ehu.es/psk31.html

[12] The technical details of PSK31 are explained in detail on the PSK31 web site.

[13] Patent "Device for the electric transmission of written characters" received 1929.

[14] See www.qsl.net/zl1bpu/FUZZY/Contents.html

[15] More details can be found at http://www.qsl.net/zl1bpu/MT63

[16] See www.qsl.net/zl1bpu/MFSK