Contents

Digital Communications

In this supplement, Steve Ford, WB8IMY, discusses the techniques involved in assembling and configuring station components for operating on the various HF and VHF digital modes. Today's digital communication choices range from keyboard based modes like classic RTTY, packet radio and PSK31, to digital voice, local high speed multimedia networks and VHF/UHF networks linked by the Internet. Related information may be found in the **Modulation** and **Digital Modes** chapters. In previous editions, this material formed Chapter 31. Unless otherwise noted, references to other chapters refer to chapters in the print version of the *ARRL Handbook*.

Amateur digital communication has always been a niche activity, but it has grown substantially over the years. From the end of World War II until the early 1980s, *radioteletype*, better known as *RTTY*, was *the* Amateur Radio digital mode. If you had visited an amateur RTTY station prior to about 1977, you probably would have seen a mechanical teletype machine, complete with rolls of yellow paper. The teletype may have been connected to the transceiver through an interface known as a *TU*, or *terminal unit*. An oscilloscope would probably have graced the layout as well, used for proper tuning of the received signal.

When affordable microprocessor technology appeared in the late 1970s, terminal units evolved as well. Some included self-contained keyboards and video displays, making the mechanical teletype obsolete. As personal computers evolved, they became perfect companions for TUs. In this configuration, the PC functioned as a "dumb terminal," displaying the received data *from* the TU and sending data *to* the TU for transmission. TUs of this era offered ultra-sharp receive filters that allowed hams to copy weak signals in the midst of interference.

In the late 1980s, conventional terminal units began to yield to sophisticated microprocessor devices known as *multimode controllers*. As the name suggests, these compact units handle several different digital modes in one package, typically RTTY, packet, AMTOR and PACTOR. Like TUs, multimode controllers are stand-alone devices that communicate with a personal computer acting as a dumb terminal. All of the heavy lifting is being done by the controller and its self-contained software known as *firmware*.

In the early 1990s, sound cards appeared for personal computers. As sound cards became more powerful, hams began to realize their potential. With the right software, a sound card could take received audio directly from the radio and translate it into digital information. The same sound card could also create various forms of digital audio modulation for transmission. The first "sound card mode" was PSK31, developed by Peter Martinez, G3PLX. In the years that followed, sound cards became more powerful and versatile. Hams responded by developing more new digital modes to take advantage of the advances.

Hardware controllers are still with us, but they are primarily used for modes like packet and PACTOR that require more processing muscle and precision timing than a typical personal computer can provide on its own. Other amateur digital modes such as D-STAR depend on specially designed transceivers that combine the radio hardware with dedicated digital processing firmware.

1 Sound Card Modes

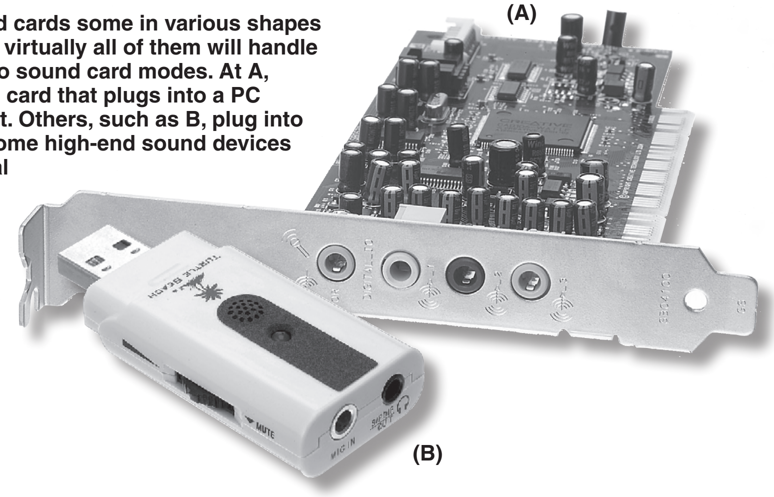
Sound card technology dominates the amateur HF digital communications world. Although the term sound *card* is commonly used, the techniques discussed in this section apply to motherboard-embedded sound chip sets, which are extremely common in computer systems today, and to external sound processing devices. We'll use the term "sound card" to refer to any of these hardware implementations. See **Fig 1**.

The sound card modes in use today include PSK, RTTY, MFSK16, Olivia, Hellschreiber and MT63. There are also sound-card applications for digital voice, discussed later in this chapter and for slow-scan TV, discussed in the **Image Communications** supplement.

The 31-baud version of PSK, known as PSK31, is by far the most popular mode for casual HF digital communication. The popularity of PSK31 notwithstanding, radioteletype (RTTY) still remains the chief HF mode for contesting and DXing. All other HF digital modes play minor

Fig 1 — Sound cards some in various shapes and sizes, but virtually all of them will handle Amateur Radio sound card modes. At A, a basic sound card that plugs into a PC expansion slot. Others, such as B, plug into a USB port. Some high-end sound devices use an internal card with the processing circuitry and an external breakout box with multiple input and output connectors.

(A)



(B)

roles, but each has characteristics that provide benefits depending on the use case. Olivia, MFSK16 and MT63, for example, provide more robust copy under poor conditions.

On VHF and above, the *WSJT* software suite is the sound card mode of choice for meteor scatter and moonbounce work. *WSJT* has also found experimental application on the HF bands. See the sidebar, "The WSJT Revolution."

Regardless of the mode in question, the sound card functions as the critical link. It is put to work as a digital-to-analog (D/A) and analog-to-digital (A/D) converter. In its A/D role, the sound card takes receiver audio and converts it to digital information. During transmission, the sound card is used as a D/A converter, taking digital information from the software application and creating a corresponding analog signal that is fed to the transceiver. (For more information on A/D and D/A converters, see the **Analog Basics** chapter.)

1.1 Which Sound Card is Best?

This is one of the most popular questions among HF digital operators. After all, the sound device is second only to the radio as the most critical link in the performance chain. Sound cards have traditional analog audio amplifiers, mixers and filters, all of which can introduce noise, distortion and crosstalk. A poor sound device will bury weak signals in noise of its own making and will potentially distort your transmit audio as well.

If you have a modest station and intend to enjoy casual chats and a bit of DXing, save your money. An inexpensive sound card, or the sound chipset that is probably on your computer's motherboard, is adequate for the task. There is little point in investing in a luxury sound card if you lack the radio or antennas to hear weak signals to begin with, or if they cannot hear you.

On the other hand, if you own the station hardware necessary to be competitive in digital DX hunting or contesting, a good sound card can give you a substantial edge. Other applications that require a lot of processing power, such as software-defined radio (SDR), require a high-performance sound card. Often vendors will offer a list of sound devices known to perform well with their equipment.

Sound cards convert analog audio signals to a set of digital samples, but this conversion from analog to digital isn't perfect, for several reasons. Here are some parameters to consider.

Sample Size

When a sound card takes a sample of the input voltage, it expresses it as a binary number with a certain number of bits. This is the *sample resolution,* or *sample size*. The sample resolution determines the number of steps between the smallest and the largest signal the card can measure. The greater the number of steps and the smaller they are, the more precise the samples will be. Larger steps introduce more *quantization noise*, so a sound card's signal-to-noise ratio is limited by the number of bits of resolution in each sample. For example, a card taking 8-bit samples measures only 256 voltage steps and cannot yield a signal-to-noise ratio better than about 49 dB. With 16-bit samples, there are 65,536 steps, and the ideal S/N rises to 98 dB.

Sample Rate

The clock that drives the A/D converter runs at a steady rate, known as the *sample rate*. As you might expect, a higher sample rate is required to accurately capture higher-frequency sounds. A waveform can be ac-

The *WSJT* Revolution

Hams routinely use meteors and the moon as radio reflectors for *meteor scatter* and *moonbounce* communication. You can read more about these activities in the chapter on **Propagation of RF Signals** and in the **Space Communications** supplement on the *Handbook* CD. For many years, meteor scatter and moonbounce required large antennas and high power RF amplifiers. CW was the mode of choice since a concentrated, narrow-bandwidth signal had the best chance to survive the journey and still be intelligible at the receiving station. That changed in 2001 when Joe Taylor, K1JT, unveiled a suite of sound card applications known simply as *WSJT*.

By using the sound card and computer as powerful digital signal processors, *WSJT* greatly reduced the station hardware requirements, making it possible for amateurs with modest stations to make meteor scatter and moonbounce contacts. Hams have also experimented with using some *WSJT* modes on the HF bands to make contacts using extremely low power. *WSJT* does not support conversational contacts with lengthy exchanges of information. Instead, the software allows for basic information exchange sufficient to meet the requirements that a contact has taken place.

WSJT is available for both *Windows* and *Linux*. It is a software suite that supports five different operating modes: *FSK441* for meteor scatter; *JT65* for moonbounce, but also being used occasionally on HF; *JT6M*, optimized for meteor scatter on 6 m; *EME Echo* for measuring the echoes of your own signal from the moon; and *CW* for moonbounce communication using 15 WPM Morse code.

The software is available for free downloading from **physics.princeton.edu/pulsar/K1JT**.

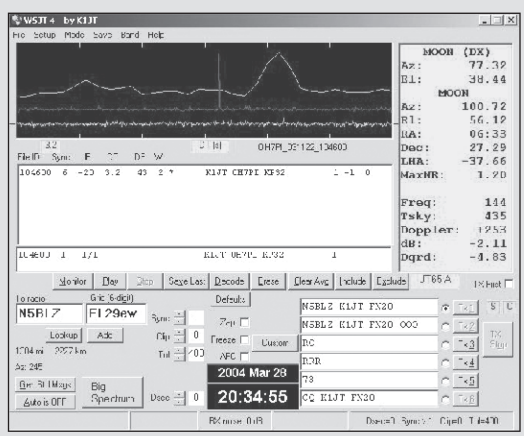


Fig A1 — WSJT software operating in the JT-65 mode.

curately captured by sampling at twice the highest frequency of interest. Energy at higher frequencies produces *aliases*, so sound cards put a low-pass filter ahead of the A/D converter, running at a cutoff frequency equal to one-half the sample rate. These filters cannot be perfect, so there's bound to be either some high frequency roll-off, or some distortion due to aliases sneaking through.

Linearity

If the sample steps aren't all exactly the same size, or the clock drifts up and down a little bit in frequency, distortion is introduced. The ideal sound card would have a perfectly linear A/D converter and a perfectly stable clock, but of course these are impossible to achieve. Good-quality sound cards do, however, have crystal-controlled oscillators as their clock source.

Sample Rate Accuracy

Even if the clock runs at a stable frequency, we won't get the desired result if it's running at the *wrong* frequency. Sample rate accuracy can be important for analog modes that aren't continuously synchronized. For these modes, one sound card is generating a signal and the other is receiving it, and the two cards are expected to be running at exactly the same sample rate. Distortion, or even loss of data, can occur if the rates are slightly different.

Remember that sound cards work by taking the analog audio signal at the input and converting it to digital information by sampling the audio signal at a very high rate, typically between 8000 and 11025 Hz for most ham applications. Sound cards can have sampling rate errors that will seriously affect weak-signal copy. For example, the laptop on which this chapter is being written has an actual sampling rate of 11098.79 Hz instead of the nominal 11025 Hz. This error not only affects the apparent frequency of an incoming tone, but can keep a program that requires a high degree of frequency stability, such as MFSK16, from maintaining consistent framing on incoming data. The result will be poor or inconsistent copy.

VoIP applications such as EchoLink and IRLP send a continuous stream of data from one sound card to another over the Internet. If the sender and receiver aren't running at the same rate, it can cause audio drop-outs, as the buffer at the receiving end becomes empty or overflows.

WSJT requires a high degree of accuracy but works around this problem by measuring the actual sample rate for sound card input and output (on some computers they are not the same) and then doing appropriate manipulations of the data. Most software doesn't have active sample error correction, however. If your software and hardware support it, you might realize better performance at a higher sample rate such as 12 kHz.

Sample rate accuracy is usually less of an issue for modes such as RTTY, PSK31, Olivia and DominoEX that synchronize the receiver with the sender frequently. In these modes, the sound card's clock is still used as the timing reference, but exact sound card timing is far less critical.

SOUND CARD STATION SETUP

Fig 2 shows a typical sound card station setup. A simplified sound card is shown here, but even the simplest of sound cards can be complicated. They can have as few as two external connections but there may be as many as twelve or more. You may find ports labeled LINE IN, MIC IN, LINE OUT, SPEAKER OUT, PCM OUT, PCM IN, JOYSTICK, FIREWIRE, S/PDIF, REAR CHANNELS or SURROUND jacks, just to name a few.

Depending on your computer, you may be able to choose your receive audio connection from either MIC INPUT or LINE INPUT. Anything else you find is not an analog audio input. If you do have a choice, use the LINE INPUT for the receive audio from your radio. Although the MIC INPUT jack can be used, it will have much more gain than you need and you may find adjustment quite critical. Some sound cards have an "advanced" option to select a 20 dB attenuator that will reduce gain and make the MIC INPUT jack easier to use.

If your only goal is to decode received signals, you need nothing more than an audio cable between the transceiver and the sound card. You should not need ground isolation for receive audio as it is at a high level and is normally not susceptible to ground loops. You may also be able to choose from several outputs that appear on the radio. Your radio may have SPEAKER, HEADPHONE, LINE OUT, RECORD, PHONE PATCH and DATA OUTPUT jacks available. These may be fixed output or variable output (using the radio's volume control). Be careful with radio's DATA OUTPUT jack — it may not work on all modes.

For the sound card transmit connection, you will have a choice of the computer's HEADPHONE OUTPUT, LINE OUTPUT, SPEAKER OUTPUT or a combination of these. The SPEAKER OUTPUT is usually the best choice as it will drive almost anything you hook to it. The SPEAKER OUTPUT has a low source impedance making it less susceptible to load current and RF. Any one of these outputs will usually work fine, provided you do not load down a line or headphone output by using very low impedance speakers or headphones to monitor computer-transmitted audio. The transmit audio connection must have full ground isolation through your interface, especially if it drives the MIC INPUT of the radio. This usually means adding an isolation transformer, as discussed in the next section.

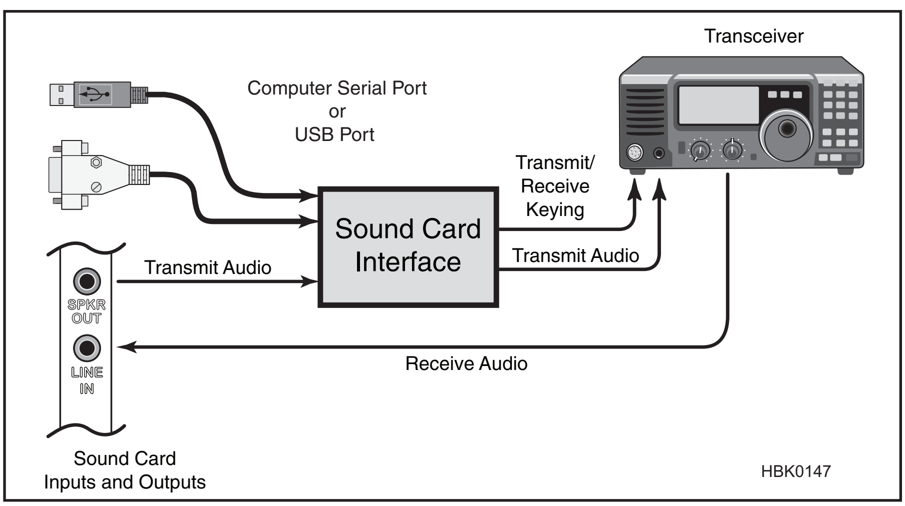


Fig 2 — A typical interface connection between a computer and a transceiver. Note that the transmit audio connects to the radio through the interface, and TR keying is provided by the computer serial port. Newer sound card interfaces are often designed to work with computer USB ports.

1.2 Sound Card Interfaces

In addition to providing audio connections between the sound card and your transceiver, you also need to provide a way for your computer to switch the radio between receive and transmit. This is where the *sound card interface* comes into play.

Commercial sound card interfaces such as the one shown in **Fig 3** match the audio levels, isolate the audio lines and provide transmit/receive switching, usually with your computer's serial (COM) or USB port. You can also make your own interface by simply isolating the audio lines and cobbling together a single-transistor switch to connect to your COM port. **Fig 4** shows some commonly used interface circuits.

In Fig 2 you'll notice that the transmit audio

Fig 3 — The MicroHam Digi Keyer is an example of a multifunction sound card interface. Some commercial devices include CW keyers, transceiver control interfaces and support for multiple transceivers.

connects to the radio through the interface. By doing so, the interface can provide isolation. Some interfaces also include a transmit audio adjustment, although this can also be accomplished at the computer, as described in the previous section.

It's important to mention that you may also be able to use the VOX function on your transceiver to automatically switch from receive to transmit when it senses the transmit audio from your sound card. This approach completely removes the need for a TR switching circuit, COM port and so on.

The weakness of this technique is that it will cause your radio to transmit when it senses *any* audio from your computer — including miscellaneous beeps, sounds or music. It's best to turn off these sounds from your computer so you don't transmit them accidentally. Another approach is to use a second sound card so that one can be used for regular computer audio applications and one dedicated to interfacing with the radio.

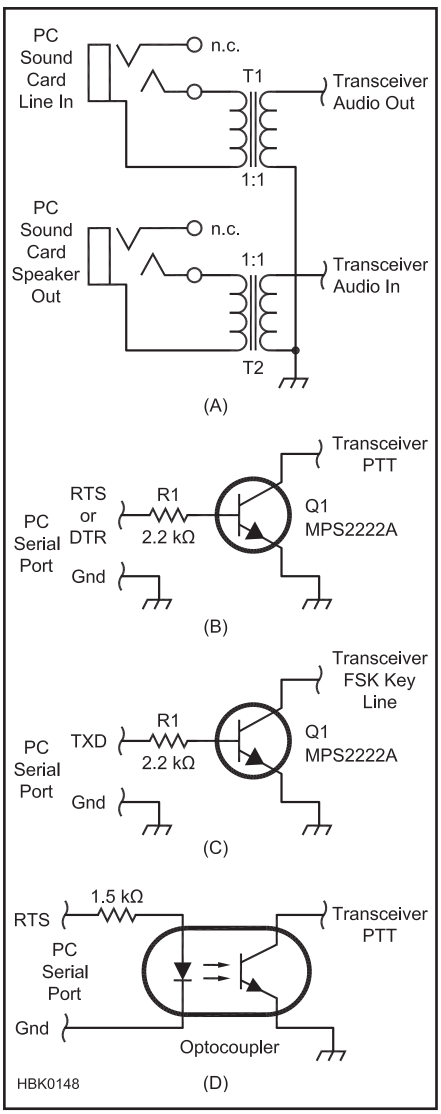


Fig 4 — Some commonly used interface circuits. At A, isolating the sound card and transceiver audio lines. T1 and T2 are 1:1 audio isolation transformers such as the RadioShack 273-1374. At B, a simple circuit to use the computer COM port to key your transceiver PTT, and at C, a similar circuit for FSK keying. Q1 is a general purpose NPN transistor (MPS2222A, 2N3904 or equiv). At D, an optocoupler can be used to provide more isolation between radio and computer. On a DB9 serial port connector: RTS, pin 7; DTR, pin 4; TxD, pin 3; GND, pin 5.

AFSK AND FSK

Most sound card modes rely on some form of frequency and/or phase-shift keying to create digitally modulated RF signals. This modulation takes place at audio frequencies with the sound card audio output applied directly to an SSB voice transceiver, either at the microphone jack or at a rear-panel accessory jack, and is called *audio frequency shift keying* (*AFSK*).

RTTY, PACTOR I and AMTOR signals can be sent using AFSK, and often are. It is also possible to transmit these modes by applying discrete binary data *directly* to the transceiver. This technique is known simply as direct *frequency-shift keying (FSK)*.

For example, each character in the Baudot RTTY code is composed of five bits. When modulated with AFSK, a "1" bit is usually represented by a 2125-Hz tone and is known as a *mark*. A "0" bit is represented by a 2295-Hz tone called a *space*. The difference between the mark and space is 170 Hz, called the *shift*. When applied to a single-sideband transceiver, this AFSK audio signal effectively generates an RF output signal that shifts back and forth between the mark and space frequencies.

A transceiver that supports FSK, however, can accept mark/space digital data directly from the computer and will use that information to automatically generate the frequency-shifting RF output. No audio signal is applied to the transceiver when operating FSK.

Is there an advantage to using AFSK or FSK? In years past, transceivers that did not support FSK operation often did not allow the use of narrow IF filtering. Those filters were reserved for CW, not the SSB voice mode used with AFSK. If you wanted to use RTTY with such a rig, you had to use AFSK and contend with the wider (2.4 kHz or so) SSB IF bandwidth, or else add an external audio filter. FSK-capable transceivers, on the other hand, allowed the RTTY operator to select narrow CW filters, reducing receive interference in crowded bands.

Many of today's transceivers offer adjustable-bandwidth digital signal processing (DSP) filters in the IF stages that can be used with any operating mode. This has effectively eliminated the FSK advantage, at least for receiving.

The appeal of FSK remains, however, when it comes to transmitting. A *properly modulated* AFSK signal is indistinguishable from FSK, but it is relatively easy to overdrive an SSB transmitter when applying an audio signal from a sound card (more on this in the next section). With FSK this is never an issue. You simply feed data from the computer to the radio; the radio does the rest.

This is why a number of RTTY operators still use FSK, and it's why transceiver manufacturers still offer FSK modes (sometimes labeled DATA or RTTY) in their products. Several sound card interfaces support FSK by providing a dedicated TTL circuit between the computer COM port, where the FSK data appears, and the transceiver FSK input. When used in this fashion, the sound card does not generate a transmit audio signal at its output. Instead, the RTTY software keys the various lines at the COM port to send the FSK data.

If the sound card interface you've chosen doesn't support FSK, you can build your own TTL interface using the circuit shown in Fig 4. This simple circuit uses a transistor that is keyed on and off by data pulses appearing on the COM port TxD pin (pin 3 on a 9-pin COM port).

It is important to note that FSK transceiver inputs can only be used for modes that are based on binary FSK, typically with 170 or 200-Hz shifts. These modes include RTTY, PACTOR I and AMTOR. You cannot use the FSK input to transmit multi-frequency or phase-shift modulated signals such as MFSK16 or PSK

1.3 Transmit Audio Levels

When applying sound card audio to a transceiver, it is critical to maintain proper levels. By overdriving the transceiver's audio input you may create a wide, distorted RF signal that will be difficult, if not impossible, to decode at the receiving end. Such an over-modulated signal will also cause considerable interference on adjacent frequencies.

As you increase the transmit audio output from your sound card, pay careful attention to the ALC indicator on your transceiver. The ALC is the automatic level control that governs the audio drive level. When you see your ALC display indicating that audio limiting is taking place, or if the display indicates that you are exceeding the ALC "range," you are feeding too much audio to the transceiver.

Monitoring the ALC by itself is not always effective. Many radios can be driven to full output without budging the ALC meter out of its "nominal" range. Some radios become decidedly nonlinear when asked to provide SSB output beyond a certain level (sometimes this nonlinearity can begin at the 50% output level). We can ignore the linearity issue to a certain extent with an SSB voice signal, but not with digital modes because the immediate result, once again, is splatter.

The simplest method to tell if your signal is clean is to ask someone to give you an evaluation on the air. For example, PSK31 programs commonly use a waterfall audio that can easily detect "dirty" signals. The splatter appears as rows of lines extending to the right and left of your primary signal, as shown in **Fig 5**. (Overdriven PSK31 signals may also have a harsh, clicking sound.)

If you are told that you are splattering, ask the other station to observe your signal as you slowly decrease the audio level from the sound card or processor. When you reach the point where the splatter disappears, you're all set.

If PSK31 is your primary digital mode, an alternative solution for *Windows* users is the PSKMeter by Software Science. This clever device samples the RF from your transceiver and *automatically* adjusts the audio output of your sound card until your signal is clean. The RF sampling port of the PSKMeter connects to your antenna coax with a T connector. The data is fed to your computer through an available COM (serial) port, or a USB port if you have a USB-to-serial converter. The PSKMeter software then analyzes the sampled signal and adjusts the master output volume of your sound card accordingly. The PSKMeter is available in kit form and can be ordered on the Web at **www.ssiserver.com/info/pskmeter**.

Whether you choose automatic or manual adjustments, don't worry if you discover that you can only generate a clean signal at, say, 50 W output. With PSK31 and most other sound card modes, the performance differential between 50 W and 100 W is inconsequential.

CONSISTENT AUDIO LEVELS

One of the perennial problems with sound devices is the fact that optimum audio levels have a tendency to differ depending on the software you are using. The mixer levels you set correctly for one application may be wildly wrong for another. And what happens when another family member stops by the computer to listen to music or play a game? They're likely to change the audio levels to whatever suits them. When it comes time to use the computer again for HF digital operating, you may get an unpleasant surprise.

The commonsense answer is to simply check and reset the audio levels before you operate. There are some software applications that will "remember" your audio settings and reapply them for you. One example is *QuickMix* for *Windows*, which is available free on the Web at **www.ptpart.co.uk/quickmix**. *QuickMix* takes a "snapshot" of your sound card settings and saves them to a file that you name according to the application (for example, "PSK31"), which can be reloaded next time you operate using that mode.

An even more elegant solution is the free *Sound Card Manager* for *Windows* by Roger Macdonald, K7QV, which you can download at **www.romacsoftware.com/Sound-Management.htm**. Whenever you start an

Computer Power

You don't need an extremely powerful computer to enjoy digital operating. You can pick up an old 850 MHz Pentium laptop on eBay (**www.ebay.com**) for less than $300. You can probably find a similar desktop system for that price and even less. Either computer will be adequate for 90% of the software you are likely to use. Serious processing power only comes into play with software-defined radios or digital voice and image applications. For those endeavors, a 2 GHz system is best for smooth performance.

Every computer needs an operating system and that's where you may encounter some sticky issues. Until 2007, the most widely used operating system was *Windows XP*. Most Amateur Radio software available today was originally written for *XP*. Much of it will also run under *Windows Vista*, *XP's* replacement, but there are no guarantees. If you are upgrading to *Vista*, or if you've purchased a computer with *Vista* already installed, you'll need to test your ham applications for compatibility. One problem that crops up frequently involves the "Help" files that so many *Windows* programs use. *Vista* does not recognize the "old" *Windows* Help format, so when you click on Help (or a portion of the *Windows* application that uses the older Help format, such as the user manual), *Vista* may present you with an error message instead. No doubt ham programmers are already adapting to this change and re-writing their applications accordingly, but it may take some time for these revisions to make it into the marketplace.

Vista also requires more CPU power to run properly. Beware of buying an underpowered new or used computer with *Windows Vista* pre-installed (and beware of installing *Vista* on an anemic machine). To obtain decent performance from *Vista*, you need a minimum 2 GHz microprocessor and 1 GB of RAM.

The good news for *Windows* users is that *XP* is likely to be around for several years to come, although Microsoft will eventually stop supporting it. *Vista* is being replaced by *Windows 7*, which offers superior performance.

Of course, you don't have to use *Windows* to enjoy HF digital. The *Linux* operating system has been gaining in popularity and you can pick up *Linux* "distributions" on the Web free of charge. For a good example, check out *Ubuntu Linux* at **www.ubuntu.com**. Mac users will find *Mac OS* applications for digital modes such as APRS and many others. See Table 1.

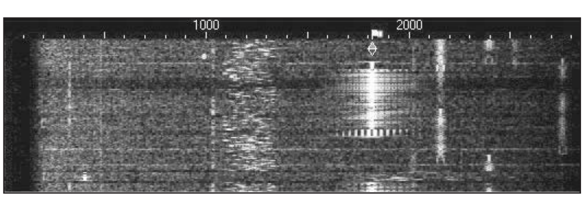


Fig 5 — The waterfall spectrum display built into popular sound card software can help detect overdriven PSK31 signals. Note the lines extending from the signal on the right, near the 2000 Hz marker, indicating splatter. The other signals are from properly adjusted transmitters.

Table 1
Sound Card Software

Windows

ALE
PCALE — **www.hflink.com**

Multimode
MixW — **www.mixw.net**
HamScope — **www.qsl.net/hamscope**
MultiPSK — **f6cte.free.fr/index_anglais.htm**
Ham Radio Deluxe — **www.ham-radio-deluxe.com**

Hellschreiber
IZ8BLY — **xoomer.virgilio.it/aporcino/Hell/index.htm**

MFSK16
IZ8BLY — **xoomer.virgilio.it/aporcino/Stream/index.htm**

PSK31
DigiPan — **www.digipan.net**
WinWarbler — **www.dxlabsuite.com/winwarbler**
W1SQLPSK — **www.faria.net/w1sql**
WinPSK — **www.moetronix.com/ae4jy/winpsk.htm**

Digital Voice
WinDRM — **www.n1su.com/windrm**

MacOS

Multimode
MultiMode — **www.blackcatsystems.com/software/multimode.html**
cocoaModem — **homepage.mac.com/chen**

Linux

Multimode
Fldigi — **www.w1hkj.com/Fldigi.html**

WINMOR
www.winlink.org/WINMOR

application (such as a piece of HF digital software), *Sound Card Manager* automatically reconfigures your sound card and then returns it to the default setting when you are done.

1.4 Sound Card Software

As you can see in **Table 1**, there is great deal of software available for sound-card-based digital modes. Some of the software applications are free, while others require registration and a fee.

There are applications dedicated to particular modes, such as *DigiPan* (PSK31) and *MMTTY* (RTTY). The trend in recent years has been to multimode programs that support many different digital modes in a single application.

Another trend has been toward *panoramic reception* where the software processes and displays all signals detected within the bandwidth of the received audio signal. The signal "signatures" are often shown within continuously scrolling *waterfall* displays like the one shown in **Fig 6**.

Panoramic reception is particularly popular among PSK31 operators because the narrow signals tend to cluster within a relatively small

Digital Communications and Public Service

In recent years, amateur digital communication has found frequent application in the realm of public service. There has been a proliferation of local and regional public-service networks, primarily using VHF packet as well as D-STAR, but three systems currently carry the lion's share of the long-haul traffic load.

Winlink 2000

Winlink 2000 is an Amateur Radio digital network with HF and VHF components. Both provide the ability to transfer e-mail to and from the Internet.

Winlink 2000 functions as a full-featured Internet-to-HF/VHF "star network" gateway system. The HF portion of the network is composed of Radio Message Server HF — *RMS HF* — stations. These stations are accessed through the use of PACTOR I, II or III and WINMOR using the multimode controller-based stations described in this chapter. On the VHF side, support is provided through *RMS Packet* stations. These packet radio stations can be accessed using common packet TNCs connected to FM voice transceivers.

Most VHF RMS Packet stations are on the air continuously. On the HF side, RMS HF stations throughout the world scan a variety of HF digital frequencies on a regular basis, listening on each frequency for about two seconds. By scanning through frequencies on several bands, these RMS HF stations can be accessed on whichever band is available at any given time.

All RMS Packet and HF stations link to a central group of redundant Common Message Servers. Since all e-mail messages coming to and from the Internet are stored on these servers and available to all RMS Packet and HF stations, Winlink 2000 users never have to designate a "home" BBS, as in the traditional packet radio world. You can send an e-mail from VHF to the Internet via a RMS Packet station in California and pick up the reply from an RMS HF station in South Africa.

In addition to text e-mail, Winlink 2000 is capable of handling file attachments such as images. However, the larger the attachment the longer it will take to send via radio. This is particularly true on VHF where most packet stations are still operating at only 1200 baud. See the Winlink 2000 Web site at **www.winlink.org**.

National Traffic System-Digital

The National Traffic System-Digital (NTSD) is based on the original Winlink structure now referred to as *Winlink Classic*. Unlike Winlink2000, which uses Internet linking between its Common Message Servers, NTSD is entirely RF based.

Most HF NTSD Mailbox Operations (MBOs) are accessed using PACTOR I or II. Designated NTSD operators in each region and local area relay messages, either between regions or to and from the local area stations. The hierarchical NTSD structure is discussed at **home.earthlink.net/~bscottmd/n_t_s_d.htm**.

Traditional packet radio is often used at the local level. In fact, NTS packet messages can be initiated and sent by any packet-capable operator. Messages for delivery are posted on cooperating NTS PBBSs (Packet Bulletin Board Systems). Messages come into these BBSs from the NTSD HF network, or from local packet networks in nearby sections or regions. In addition, any NTS voice messages that might not have been picked up on a voice net can be transcribed to text and posted to an NTS PBBS.

Global ALE High Frequency Network

Global ALE High Frequency Network (HFN) is an international Amateur Radio Service organization of ham operators dedicated to emergency/relief radio communications. The main purpose of the network is to provide efficient communications to remote areas of the world using amateur Automatic Link Establishment (ALE) as described in this chapter. The network supports direct station-to-station emergency text messaging as well as the ability to port text messages to and from the Internet. See **hflink.com/emcomm/.**

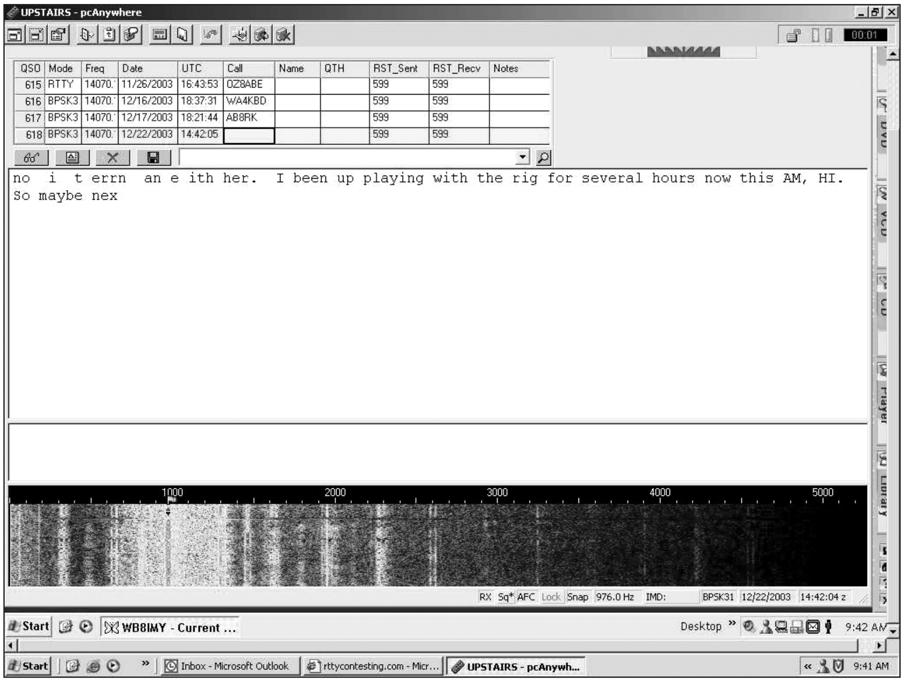


Fig 6 —Multimode sound-card applications typically support a variety of modes such as RTTY, PSK31, Olivia and MFSK16. Sent and received text appear in the white box. At the bottom of the screen is a waterfall tuning display. Tune the transceiver VFO to the calling frequency and then select stations to work by clicking on the signal trace in the display.

range of frequencies. By using panoramic reception, an operator can simply click the mouse cursor on one signal trace in the waterfall after another, decoding each one in turn.

The weakness of panoramic reception appears when wide IF filtering is used to display as many signals as possible. The automatic gain control (AGC) circuit in the receiver is acting on everything within that bandwidth, working hard to raise or lower the overall gain according to the overall signal strength. That's fine if all the signals are approximately the same strength, but if a very strong signal appears within the bandwidth, the AGC will *reduce* the gain to compensate. The result will be that many of the signals in the waterfall display will suddenly vanish, or become very weak, as the AGC drops the receiver gain. In cases where an extremely strong signal appears, *all* signals except the rock crusher may disappear completely.

The alternative is to use narrower IF or audio filter settings. These will greatly reduce the waterfall display width, but they will also remove or reduce strong nearby signals.

2 Packet Radio

Although the heyday of traditional packet radio has passed, it still survives in the form of networks devoted to specific applications such as emergency communication (see the sidebar "Digital Communications and Public Service") and DX spotting. A variant of packet radio also finds popular use in the Automatic Packet/Position Reporting System (APRS), addressed in a later section.

2.1 The Packet TNC

In the beginning, there was *X.25*, a protocol for wide-area digital networks that typically communicated over telephone lines. X.25 works by chopping data into strictly defined *packets* or *frames* of information. Each packet is sent to the destination device where it is checked for errors. If errors are discovered, the packet must be sent again to ensure that received data is 100% error free.

In the early 1980s, amateurs began adapting X.25 for over-the-air digital communications. The result was *AX.25*. The new AX.25 protocol worked in much the same way, although it identified each message by sender and destination station call signs and added a Secondary Station ID (SSID) in a range from 0 through 15.

To create and decode these AX.25 packets, hams invented the *terminal node controller*, or *TNC*. TNCs do much more than assemble and disassemble data; they are miniature computers unto themselves. A TNC is also programmed to work within a radio network where there may be other competing signals. For example, to maximize the throughput for everyone on the same frequency, a TNC is designed to detect the presence of other data signals. If it has a packet to send, but detects a signal on the frequency, it will wait until the frequency is clear. TNCs also have a variety of user adjustments and other features, such as a mailbox function to store messages when the operator is away.

Regardless of the changes in packet radio, the TNC is still a vital component. In essence, a TNC functions as a "radio modem." It acts the middleman between your radio and your computer. The TNC takes data from your computer, creates AX.25 packets and then transforms the AX.25 formatted data into audio signals for transmission by the radio. Working in reverse, the TNC demodulates the received audio, changes it back into data, disassembles the AX.25 packets and sends the result to the computer.

For 300 and 1200-baud applications, TNCs create signals for transmission using AFSK. The most common is 1200 baud packet, used primarily at VHF. When creating a 1200-baud signal, a *mark* or 1 bit is represented by a frequency of 1200 Hz. A *space* or 0 bit is represented by a frequency of 2200 Hz. The transition between each successive mark or space waveform happens at a rate of 1200 baud. The frequencies of 1200 and 2200 Hz fit within the standard narrowband FM audio passband used for voice, so that AFSK is accomplished by simply generating 1200 and 2200 Hz tones and feeding them into the microphone input of a standard FM voice transmitter.

FSK, not AFSK, is used for 9600 baud packet. The data signal is filtered and encoded and then applied directly to the FM transmitter, after the microphone amplifier stage, through a dedicated port.

A block diagram of a typical TNC is shown in **Fig 7**. You'll note that it has a serial interface connecting to a "terminal." Most commonly, the terminal is a full-fledged computer running terminal software. Data flows to the computer and vice versa via this interface. At the heart of the TNC is the microprocessor and the attendant *High-level Data Link Controller*, or *HDLC*. The microprocessor is the brain of the unit, but the HDLC is responsible for assembling and disassembling the packets. The modem is simply that — a modulator (changing data to audio tones) and demodulator (changing audio tones back to data).

You can still find TNCs for sale from several manufacturers. There are also several transceivers with packet TNCs built in.

Another solution is a software TNC known

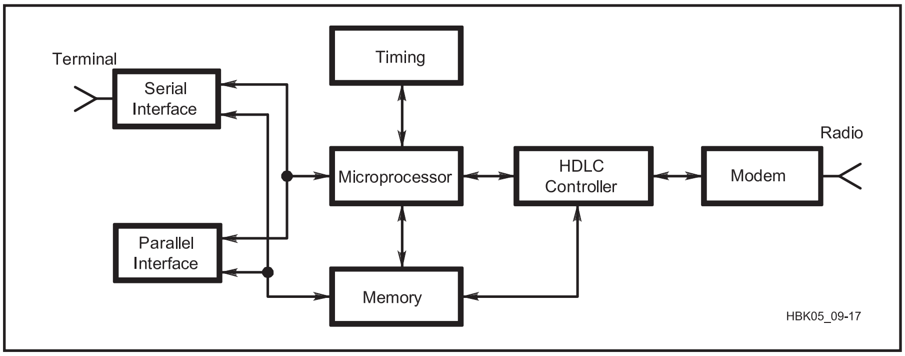


Fig 7 — A block diagram of a typical packet radio TNC.

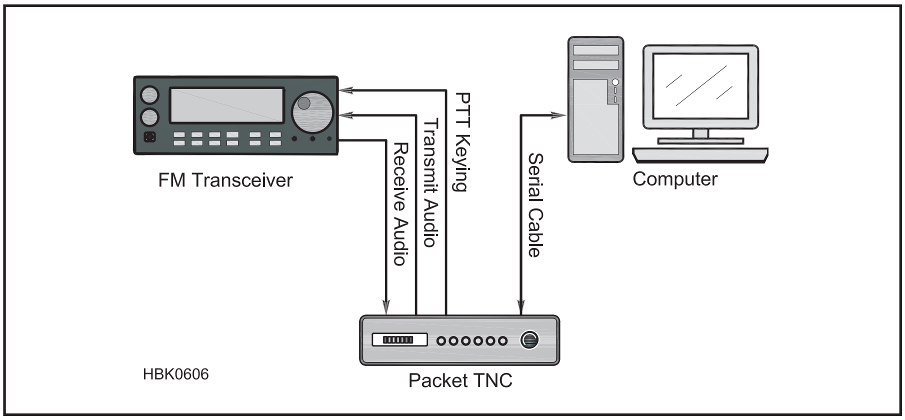


Fig 8 — A diagram of a packet radio station built around a hardware TNC.

as the *AGW Packet Engine*, or *AGWPE*, developed by George Rossopoylos, SV2AGW. The software is downloadable at **www.sv2agw.com/downloads**. With *AGWPE* and a sound card interface as described in this chapter, it is possible to operate packet radio without a hardware TNC — assuming that your primary packet software application includes the ability to interface with *AGWPE*.

2.2 Talking to a TNC

All hardware TNCs are functionally the same. When you buy a stand-alone TNC it usually includes a cable for connecting it to the radio, but you'll have to attach the appropriate mic and speaker jack connectors for your radio. You'll also have to furnish the cable that connects the TNC to your computer at the COM port. In most cases this is an RS-232 serial cable. See **Fig 8** for a typical setup. Most ham TNCs have yet to migrate to USB at the time of this writing. TNCs integrated into radios may need a cable between the radio's TNC port and a computer.

TNC manufacturers often supply the software necessary to communicate with the TNC, but any terminal program will work. You'll need to start the software and specify the COM port you'll be using, and set the baud and data parameters for that port. Refer to the manual for the specific program you've chosen. The baud rate of your computer must match the baud rate of your TNC. Some TNCs will automatically set their baud rate to match the computer; other TNCs have software commands or switches that determine the baud rate. Again, you'll need to refer to your manual for specific instructions. When setting the data parameters, 8-N-1 is normally used: 8 data bits, no parity, and 1 stop bit.

When you switch on the TNC, you should see some sort of "greeting" text on your screen. That's the first sign that all is well. If you see a bunch of gibberish, it means that the parameters of the TNC and computer don't agree and you'll have to make adjustments.

2.3 TNCs and Radios

Tones for 1200 baud can be fed directly into the microphone input of any VHF FM voice transceiver with an appropriate cable. Packet at 9600 baud requires an FM transceiver with a dedicated 9600-baud packet input, as noted earlier.

If you opt to craft your own cable, check your transceiver manual for the wiring diagram of the microphone jack. In most cases, there are separate connections for the audio input and the push-to-talk (PTT) line. (The TNC grounds the PTT line to key your transceiver.) Some transceivers also make receive audio available at the microphone jack for use with speaker/microphone combos. You can use this line to feed audio to the TNC. If it isn't available, you will have to make a separate connection to the transceiver's external speaker jack.

When you are setting up your TNC, be careful about the transmit audio level. Too high a level will create distorted signals that won't be decodable at the receiving station.

An easy way to check your transmitted signal is to use the TNC *calibrate* function. Get to the command mode (CONTROL-C) and enter **cmd: CALIBRATE**. Listen to your transmitted signal with another rig and raise the audio level from the TNC until the received volume seems to stop increasing. Now reduce the audio from the TNC until you can just hear a volume decrease in the receiver. Reduce it a bit more and you're done. Exit the calibrate function.

Some TNCs have an audio output adjustment on the board, some have an adjustment accessible through a hole in the side of the unit, and some have two fixed output levels selectable with a jumper. If one of these does not work, you may have to open up the transceiver and find the mic gain control. If this is necessary, be sure you adjust the mic gain control and *not* the deviation control (which can change the bandwidth of your signal).

2.4 TNC Timing

The TNC's TXDELAY parameter specifies the delay interval between the time the TNC keys the radio and the time it starts sending data. Normally 300-400 ms are adequate, but some transceivers take longer to settle after the keying line is triggered. If you have a problem being heard and your audio seems normal, try increasing TXDELAY to 400-600 ms.

On a busy network, packets and packet acknowledgements fly back and forth at a furious rate. One way to keep interference to a minimum is to manipulate the RESP and DWAIT parameters in conjunction with PERSIST and SLOTTIME to allow staggered transmissions.

RESP is the time delay between reception of a packet and transmission of an acknowedgement. DWAIT sets the delay between the time when activity is last heard on the channel and the moment your radio transmits. You should set values of RESP and DWAIT to the values recommended in your area (check with the person managing the local network or PBBS). Your TNC probably accepts a value in "counts" rather than in milliseconds, so don't forget to convert by the proper value in order to arrive at the correct timing value in milliseconds. For example, if you have been asked to set DWAIT to 600 ms and the units of DWAIT for your TNC are 10 ms per count, then you would command DWAIT = 60.

Most TNCs contain commands called PERSIST and SLOTTIME, which help enormously in avoiding interference. PERSIST sets the probability that a packet will be transmitted during a given time interval called a SLOTTIME. The parameter SLOTTIME governs the interval between transmission timing "slots." Initially, PERSIST should be set to approximately 64 and SLOTTIME to a value of about 10, which is equivalent to 100 ms.

FRACK (frame acknowledgement) should be set to 6 and RETRY to 10. FRACK sets the number of seconds between retries and RETRY sets the number of times your TNC will try to send a packet and gain acknowledgement of it before it gives up and disconnects

2.5 Monitoring

Start by listening to an active packet frequency in your area. With the radio cable connected, turn on your radio and increase the receiver volume partway. Some TNCs include an LED indicator that shows that the TNC is receiving audio. Turn up the squelch control on the radio until the LED is extinguished. Tune the rig to any odd numbered frequency between 144.91 and 145.09, or between 145.61 and 145.79 MHz, and set the rig for simplex operation. Your best bet may be to search for a DX PacketCluster, or try monitoring APRS activity on 144.39 MHz. When you hear the buzzing packet signals and see text on your screen, you'll know you've hit the jackpot.

Depending on the type of activity you are monitoring, you may see what appears to be nonsense. If you are monitoring APRS, you'll see strings of numbers. These are latitude/longitude position reports. On PacketClusters, you'll see DX call signs and frequencies.

2.6 "Connected" vs "Unconnected"

When discussing TNCs and networks, it is important to understand the difference between connected and unconnected communication.

If you are simply monitoring local packet transmissions, your TNC is in an *unconnected* state. What you see is what you get. If a signal is garbled by noise or interference, you'll see nothing on your screen (unless you've enabled the PASSALL function, in which case you'll see gibberish). If you transmit an unconnected packet, the signal simply leaves your antenna destined for nowhere in particular. Some stations may decode it, some may not.

When your TNC is operating in a *connected* state, everything changes. When you are connected, your station is linked to another station in a "virtual" sense. In a connected state, every packet you send is intended specifically for the receiving station (even though others can see it).

When your TNC transmits a packet, it starts a countdown clock. If the clock reaches zero before your TNC receives an acknowledgement (known as an *ACK*) that the packet arrived without errors, it will send the same packet again. When the packet is finally acknowledged, the TNC will send the next packet. And so it goes, one packet after another. The operator at the other station may also be sending packets to you since this communication process can flow in both directions simultaneously.

The big advantage of the connected state is that data is delivered error-free. One packet station can connect to another directly, or through a series of relaying stations. Error-free can be a disadvantage, too. Specifically, a connected state works best when signals are strong and interference is minimal. Remember that if too many packets are lost — by either not arriving at all or arriving with errors — the link will fail. That's why AX.25 packet radio tends not to work well on the HF bands. With all the noise, fading and interference, packets are often obliterated enroute.

Unconnected packets are ideal for applications where you are transmitting essentially the same information over and over. Since unconnected packets can be decoded by any station, they are an excellent means of disseminating noncritical data (data that doesn't need guaranteed error-free delivery) throughout a given area. If a station fails to decode one packet, it merely waits for the next one. APRS uses exactly this approach.

3 The Automatic Packet/Position Reporting System (APRS)

The Automatic Packet/Position Reporting System, better known as *APRS* (**aprs.org**), is the brainchild of Bob Bruninga, WB4APR. In fact, APRS is a trademark registered by WB4APR. The original concept behind APRS involved tracking moving objects, and that's still its primary use today. However, APRS has been evolving to become an amateur network for other applications such as short text messaging, either from ham to ham, or between hams and non-hams through APRS Internet gateways.

Mobile APRS stations communicate their positions based on data provided by onboard Global Positioning System (GPS) receivers. The GPS receivers are attached to either packet radio TNCs or simplified packet devices known as *position encoders*, which in turn are connected to transceivers (see **Fig 9**). At receiving stations, various APRS software packages decode the position information and display the results as icons on computer-generated maps such as the one shown in **Fig 10**. When a station moves and transmits a new

Fig 9 — A 2 meter FM handheld transceiver attached to an APRS position encoder.

Fig 10 — A snapshot of APRS activity using *UI-View* software. Note the icons representing various mobile and fixed stations.

position, the icon moves as well.

Virtually all APRS activity takes place today on 144.39 MHz using 1200-baud packet TNCs and ordinary FM voice transceivers. In areas where the APRS network is particularly active, you may hear traffic on 445.925 MHz as well. There is also some activity on HF.

3.1 Setting Up an APRS Station

If you own a 2 meter FM voice transceiver, you already have the primary component of your APRS station. Tune your radio to 144.39 MHz and listen for packet transmissions. If you hear them, it means you have APRS activity in your area.

To decode APRS packets, you'll need a TNC, and it doesn't necessarily have to be "APRS compatible." APRS compatibility is only a factor if you wish to connect the TNC to a GPS receiver to transmit position data.

If all you want to do is monitor APRS activity, you do *not* need a GPS receiver. If you want to participate in the local APRS network from a fixed (non-moving) station such as your home, you still do not need a GPS receiver. Just determine your home latitude/longitude coordinates and you can use them to establish the location of your home station on the network. There are numerous sites on the Internet that will convert your home address to a correct latitude and longitude.

The only APRS station that requires a GPS receiver is a *moving* station. The good news is that almost any GPS receiver will do the job. It does not have to be elaborate or expensive. The only requirement of an APRS-compatible GPS receiver is that it provide data output in *NMEA* (National Maritime Electronics Association) format. Note that many GPS receivers advertise the fact that they provide data output, but some do it in a proprietary format, not NMEA — check the manual. See **Fig 11** for a diagram of a typical APRS station with a GPS receiver.

APRS-compatible TNCs and tracking devices have standardized on the NMEA 0183 protocol. They expect data from the GPS receiver to be in that format so they can extract the necessary information and convert it into AX.25 packets for transmission. If the data from the GPS receiver is incompatible, the TNC or position encoder won't be able to make sense of it.

The final component of your APRS station is software. You'll need software to display the positions of APRS stations, along with other information contained in their transmissions. APRS software is also essential if you want to communicate over the APRS network from a fixed or portable station. Note, however, that APRS software is *not* necessary for mobile stations that wish to merely transmit APRS beacons for tracking purposes. That function is carried out automatically using the GPS receiver and APRS-compatible TNC or tracking device; it does not depend on software.

The most popular APRS *Windows* program is *UI-View*. *UI-View* was created by the late Roger Barker, G4IDE. You'll find it on the Web at **www.ui-view.org**. The 16-bit version is free for downloading. To use the 32-bit registered version, hams are asked to donate to their local cancer charities. Details are available on the *UI-View* Web site. For the Macintosh, there is *MacAPRS* available from **www.winaprs.com**. For *Linux* there is *Xastir,* available for download at **www.xastir.org**.

APRS software, regardless of the operating system, is designed to talk to the packet TNC, processing the incoming APRS data and creating icons on your computer screen. The application also uses the TNC to transmit APRS data. This means that the software and the TNC must communicate with each other at the same baud rate. Every APRS application has a setup menu to program the correct parameters for various TNCs.

Depending on the software, there may be other features such as logging or messaging. APRS software changes rapidly, so check the help file or manual for specific instructions.

3.2 Maps and APRS

No matter which APRS software you choose, one absolutely critical aspect is the mapping function itself. To get the most from APRS, your software maps must be as comprehensive as possible, preferably with the ability to show detail down to street level.

Downloadable APRS software applications generally do not come with detailed maps. Detailed map files are numerous and large, impractical to bundle with every APRS program. Instead, most applications are designed to import user-created custom maps, or to work with existing commercial mapping programs such as Microsoft *Streets*, Delorme *Street Atlas* and *Precision Mapping*.

UI-View, for example, has the ability to automatically load and display maps from *Precision Mapping*. You must purchase and install *Precision Mapping* on your PC, then download and install a small *Precision Mapping* "server" application into *UI-View*.

Each APRS transmission includes characters that define the type of map icon that will be displayed at the receiving end. If you are operating a fixed station, your APRS software will allow you to choose your icon. If you are a mobile station using a traditional TNC, you'll need to define your chosen icon in your beacon statement. APRS-compatible TNCs give you the ability to do this. APRS position encoders also allow you to choose your icon when you program the unit. Your mobile icon might be a car, boat or airplane.

3.3 APRS Position Encoders

You can create a mobile APRS station with a VHF FM transceiver, a TNC and a GPS receiver. Wire everything together, connect an

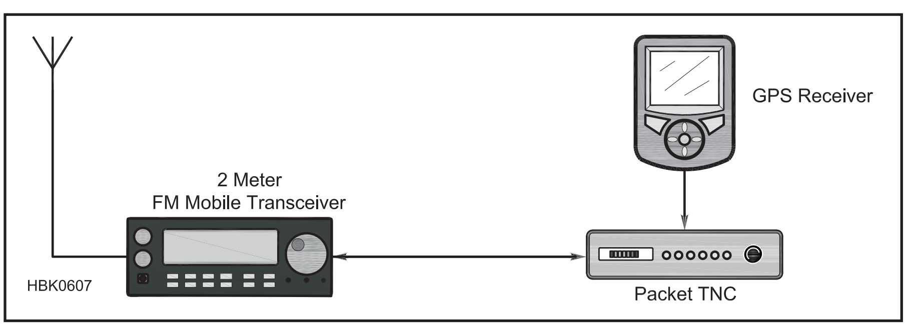


Fig 11 — Diagram of an APRS station. This configuration includes a GPS receiver, although the receiver is only required if the station is mobile. In mobile applications, a position encoder can be substituted for the packet TNC.

Fig 12 — Some APRS position encoders, such as this RPC Electronics RTrak, combine the GPS receiver, packet circuitry and 2 meter transmitter into a single device.

antenna and dc power and you're set. For hams on the go, however, it's common to replace the full-fledged TNC with an APRS *position encoder*, also referred to as an APRS *tracker*. A position encoder is a compact device designed for one purpose: to receive data from the GPS receiver, assemble APRS packets from the data and create modulated signals for use by the transmitter. Some position encoders include GPS receivers in their designs. You'll even find position encoders that are complete packages incorporating tiny GPS receivers and low power FM transmitters (**Fig 12**).

To use a position encoder you must program it the same way that you initially program a TNC. Like TNCs, position encoders connect to computer serial or USB ports for programming and most come with their own programming software. You must enter your call sign and other information such as your beacon interval (how often you want the position encoder to transmit your position). Most position encoders allow you to set the beacon interval to a certain amount of time (say, every two minutes). Some position encoders can be configured to transmit position beacons after a certain distance (every mile), or whenever the vehicle turns a corner.

3.4 APRS Networking Tips

One of the key features of APRS is that while it uses AX.25 to transport its messages, it essentially ignores all the AX.25 connection-oriented baggage. Unlike traditional packet radio, APRS stations do not establish "connections" with each other. Instead, APRS packets are sent to no one in particular, meaning to *everyone*.

Every APRS station has the ability to function as a digital repeater, or *digipeater*. So, if it receives a packet, it will retransmit the packet to others. As other digipeaters decode the same packet, they will also retransmit and spread it further. This is known as *flooding* and is illustrated in **Fig 13**.

As an APRS user, you can set up your station to address its packets through specific digipeaters according to their call signs. But when you're traveling, how do you determine which digipeaters you should use?

PATHS AND ALIASES

In the packet world, nodes and digipeaters can have *aliases*. A digipeater call sign may be W1AW-1, but it can also carry an alias, using the MYALIAS command in the digipeater TNC. Perhaps the digipeater alias would be NEWNG (meaning the ARRL HQ home town of Newington). You can route packets through the digipeater by addressing them to W1AW-1, or simply by addressing them to NEWNG. Any station that is set up to respond to an alias is capable of handling your packets automatically, even if you don't know its call sign.

Unlike typical packet use of aliases in which a given single station has a specific alias, APRS specifies standard digipeater aliases that nearly all stations use. This means that you can travel anywhere in the country and still participate in the APRS network without knowing digipeater call signs. (Otherwise, you'd have to reconfigure your TNC whenever you moved from one area to another.)

The most common APRS digipeater alias is *WIDEn-N*. The letters "n" and "N" represent numbers. The first (left-most) "n" designates how many WIDE digipeaters will relay your packets, assuming they can receive them. WIDE2, for instance, is the same as saying that you want your packets relayed through

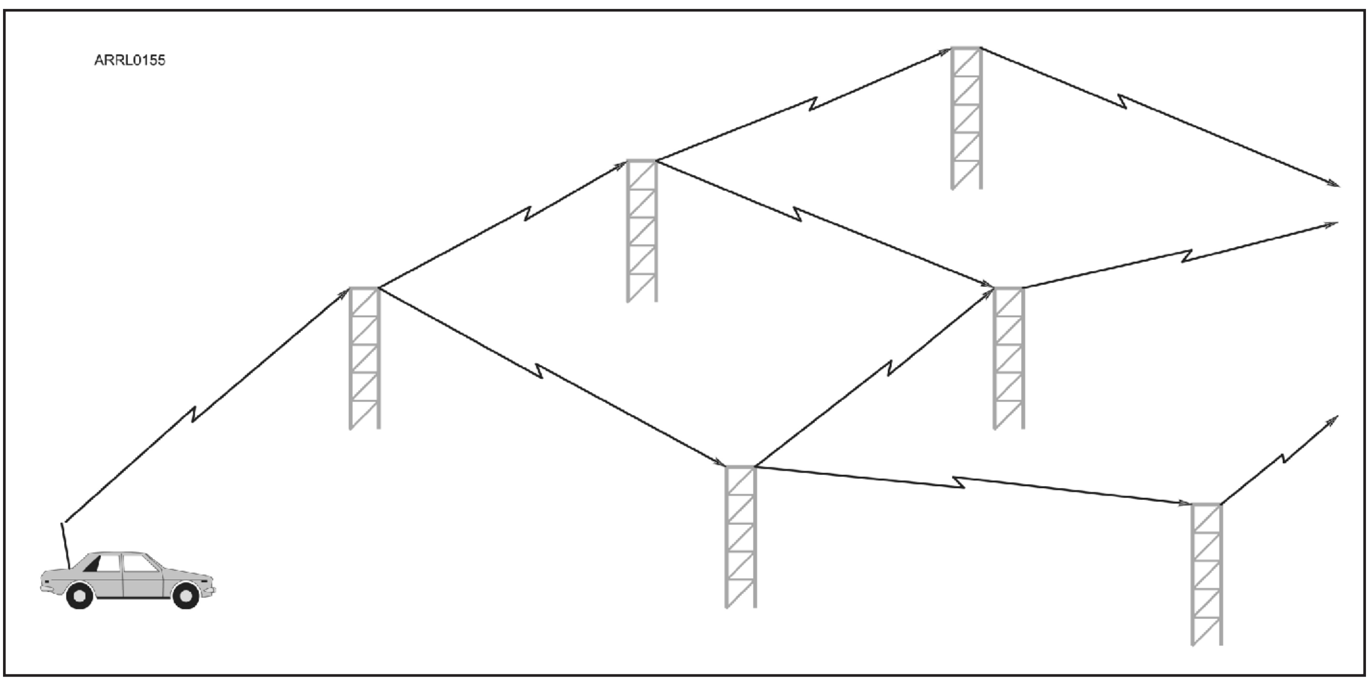


Fig 13 — In this example, an APRS packet is transmitted by a mobile station and is retransmitted by a nearby digipeater. Depending on how the mobile operator configured his TNC or tracker's path, the packet will be picked up and repeated by several other digipeaters. This is known as flooding.

two WIDE digipeaters. The second "N" is the *Secondary Station Identifier* (SSID). The SSID is used in APRS networks as a means of limiting how often (and how far) a packet can be repeated.

Here's how it works. Each time your packet traverses a WIDEn-N digipeater, the digipeater subtracts 1 from the SSID as it retransmits. The next digipeater deducts 1 and so on until the SSID reaches zero, at which time the packet will not be repeated again. This has the effect of limiting the flood radius. See **Fig 14**.

When you configure a TNC or position encoder for use with APRS, you can use these aliases to set up the paths for the beacon packets you'll be transmitting. In most devices this is accomplished with the UNPROTO parameter, sometimes simply referred to as the "Path." If you are a fixed station (a station at home, for instance), set your path as **WIDE2-2** (or with a traditional TNC UNPROTO statement, set it to APRS VIA WIDE2-2). This designates that your reports will be relayed by *two* WIDEn-N digipeaters and limits the spread beyond those repeaters to just two retransmissions. Set your TNC to beacon once every 30 minutes. That's sufficient for a fixed station.

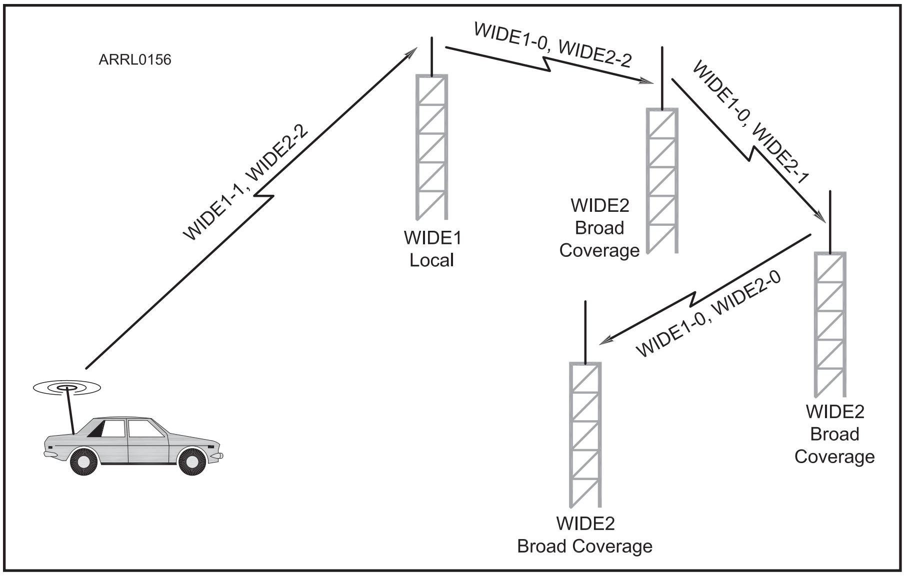


Fig 14 — By using the WIDEn-N system, we can limit packet flooding in a local network and greatly reduce congestion. The mobile station in this example has his path set as WIDE1-1,WIDE2-2. Notice how his packet propagates through the network and how the SSID number is reduced by one each time the packet is repeated through a digipeater with a corresponding alias. When it reaches the third WIDE2 digipeater, the counters all reach zero and digipeating stops.

If you are running APRS from a car, try **WIDE1-1,WIDE2-1** (or APRS VIA WIDE1-1,WIDE2-1). WIDE1-1 ensures that your packet will be picked up by at least one local digipeater or a home station acting as a fill-in digipeater and relayed at least once. WIDE2-1 gets your packet to another, presumably wider-coverage digipeater, but limits the retransmissions beyond this point. It's wasteful of the network to set up wide coverage for a station that is rapidly changing its position anyway. Mobile stations that are in motion should also limit their beacon rate to once every 60 seconds, or once per mile, whichever comes first.

Never invoke extremely wide coverage, such as a WIDE5-5 path, unless you are way out in the hinterlands and need every relaying station available to get your packets into the network.

Another popular alias is the "single state" — SSn-N — to limit the spread of your packets to a specific state or area within a state. To keep packets within the state of Connecticut, for example, you could use the SSn-N alias in a path statement, like this: CT1-1, CT2-2. This path assures that local Connecticut stations (CT1-1) will repeat the packets, and that broad-coverage stations (CT2-2) will relay them throughout a large portion of the state. APRS digipeaters outside Connecticut, however, will not respond to these packets because they won't recognize the CT2-2 alias (although they will recognize an alias for their state). The packets will still be heard across state borders but will not be digipeated or add to the packet activity in a neighboring state.

4 PACTOR

PACTOR is a type of HF digital communication that, unlike many other HF modes, offers error-free text and file transfers. PACTOR communication resembles packet radio in that it establishes a "link" between two stations. Data is sent in discrete frames and each frame is acknowledged by the receiving station. This rapid back-and-forth exchange creates PACTOR's distinctive *chirp-chirp-chirp* signal. Through robust coding and sophisticated modulation techniques, PACTOR is often able to maintain the link even in the face of significant noise, fading and interference.

PACTOR I, introduced in the early 1990s, has been largely superseded by PACTOR II,

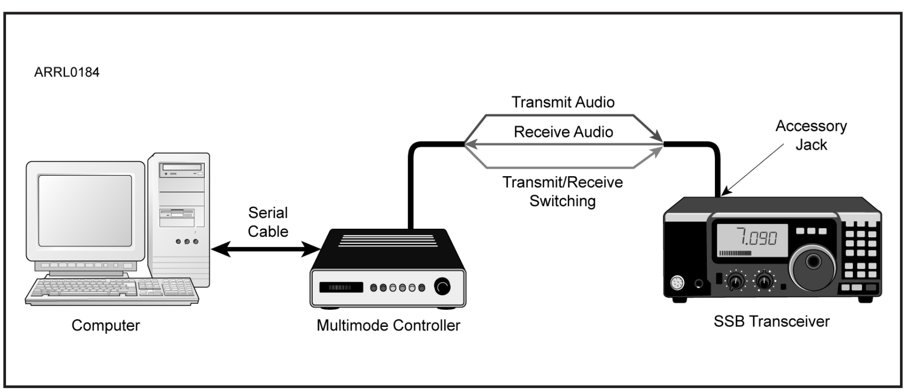


Fig 15 — A PACTOR station with a multimode controller. For PACTOR II and III operation, the controller must be a model manufactured by the SCS Corporation.

which provides faster throughput in difficult conditions. Both PACTOR I and II generate signals that occupy 500-Hz bandwidths. PACTOR III offers even more efficient communication, but its signal occupies a bandwidth in excess of 2000 Hz.

To set up a PACTOR station, you must purchase a stand-alone multimode controller that includes the PACTOR mode. As mentioned at the beginning of this chapter, multimode controllers are microprocessor-based devices that support several different digital modes in a single package. A typical multimode controller might include PACTOR, packet, RTTY and more. Controllers manufactured by Kantronics (**www.kantronics.com**) and Timewave (**www.timewave.com**) include PACTOR I. For PACTOR II or III, you must purchase a controller made by Special Communications Systems (**www.scs-ptc.com**), the original developers of PACTOR.

Fig 15 illustrates a typical PACTOR station built around a multimode controller. The computer is functioning only as a dumb terminal for the controller in this application, so it does not have to be particularly powerful. All that is required is basic terminal software as described in the section on packet TNCs. More sophisticated software applications can provide a smoother, easy-to-use interface, but these require more capable computers.

A PACTOR station requires an SSB transceiver capable of switching from transmit to receive within approximately 30 ms. Most modern HF transceivers can meet this requirement, but not all. *QST* magazine Product Reviews test this specification for all HF transceivers. The transmit audio supplied by the controller can be fed to the microphone or accessory jack to operate AFSK, which is the standard procedure, although PACTOR I can also operate using FSK.

The most popular use of PACTOR on amateur frequencies today is within the *Winlink 2000* network. PACTOR is also widely used as the HF digital arm of the National Traffic System-Digital (NTSD). See the sidebar "Digital Communications and Public Service."

5 High Speed Multimedia (HSMM)

Wireless networking using IEEE 802.11 standards has seen explosive growth during the last several years. Coffee shops, fast-food restaurants, hotels, airports and many other high-traffic locations now include wireless Internet (WiFi) hotspots. Some hotspots offer free Internet access while other charge an access fee, payable with your credit card.

Wireless networks are also popular at home. Establishing a home network is as simple as installing a *wireless router* to manage the data flow from the broadband Internet connection. The router allows one or more traditional desktop computers to tap the broadband connection through wired (usually Ethernet) access while simultaneously making the Internet available wirelessly to one or more laptops.

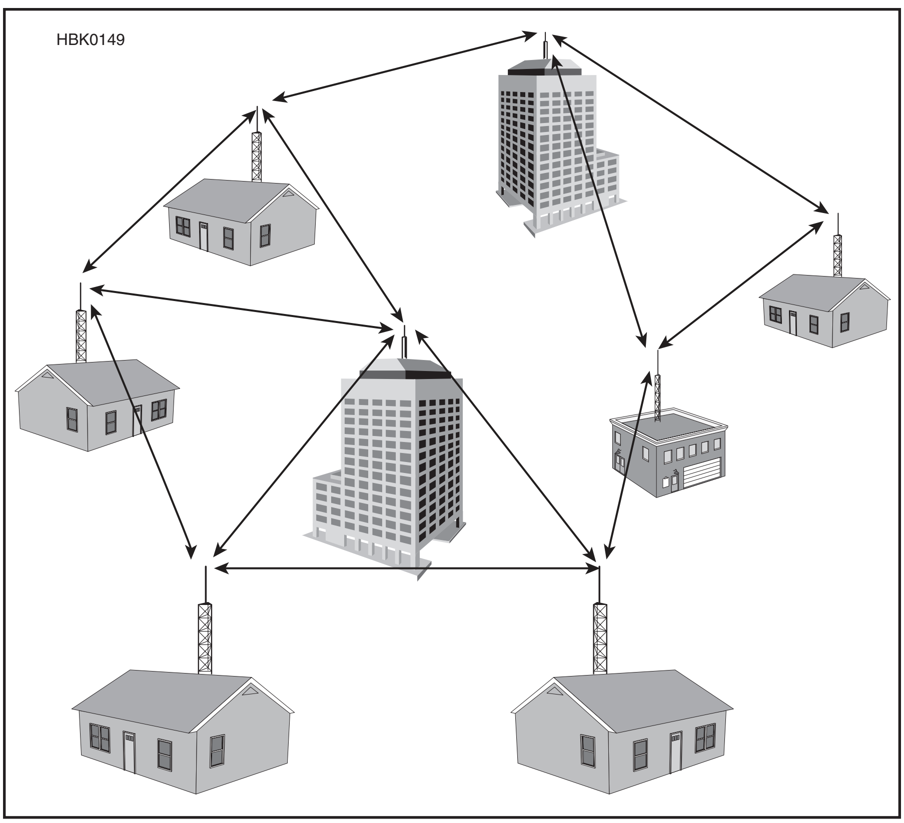


Fig 16 — An HSMM Mesh Network is highly decentralized. Individual mesh stations (nodes) act as repeaters to transmit data from nearby nodes. If one node drops out of the network, the other nodes will automatically re-route the traffic.

All these home networking devices — routers, wireless access cards, and so on — are unlicensed FCC Part 15-regulated transceivers with RF outputs measured in milliwatts. A number of their channel frequencies overlap two Amateur Radio bands: 2.4 and 5.8 GHz. This means that hams can put them to work as "transceivers" under our Part 97 rules. See **Table 2** for specific frequencies.

By using consumer-grade routers, low-loss coaxial cable and gain antennas, hams can quickly establish high-speed, long-distance wireless networks on these 2.4 or 5.8 GHz frequencies. This type of operating is referred to as *high speed multimedia*, or *HSMM*.

With HSMM amateurs have the opportunity to operate several different modes at the same time, and usually do. HSMM is generally IP-based, and given enough bandwidth, radio amateurs have the capability to do the same things with HSMM that are done on the Internet.

• *Audio*: This is technically digital voice, since it is two-way voice over an IP (VoIP) network similar to EchoLink and IRLP networks used to link many Amateur Radio repeaters over the Internet. (VoIP is covered in a later section.)

• *Video*: Motion and color video modes, are called amateur digital video (ADV). This is to distinguish it from digital amateur television (DATV). DATV uses hardware digital coder-decoders (CODEC) to achieve relatively high-definition video similar to *entertainment quality TV*. The usual practice in

Table 2
Wireless Network Frequencies in Amateur Bands

Channel	*Center Frequency*	*Channel*	*Center Frequency*
1	2.412 GHz	132	5.660 GHz
2	2.417 GHz	136	5.680 GHz
3	2.422 GHz	140	5.700 GHz
4	2.427 GHz	149	5.745 GHz
5	2.432 GHz	153	5.765 GHz
6	2.437 GHz	157	5.785 GHz
		161	5.805 GHz
		165	5.825 GHz

HSMM radio is to use a far less expensive (often free) PC-based software video CODEC to achieve *video communications quality* signals in much smaller bandwidths.

• *Text:* Text exchanges via a keyboard are often used in HSMM radio, but they are similarly called by their Internet or packet radio name: *Chat mode*, and if a server is available on the network, e-mail can also be exchanged.

• *Image*: Image file transfers using file transfer protocol (FTP) can also be done, just as on the Internet.

• *Motion video*: FTP of MPEG files can provide one-way video streaming of short video clips.

• *Remote control*: Individual devices or even complete stations can be remotely controlled.

• *Mesh networking*: A mesh is a wireless cooperative communication infrastructure in which each station functions as a relay, allowing the entire network to cover a large area. See **Fig 16**. This type of infrastructure can be decentralized (with no central server) or centrally managed (with a central server). Both are relatively inexpensive and very reliable. Individual mesh stations (nodes) act as repeaters to transmit data from nearby nodes. The reliability comes from the fact that each node is connected to several other nodes. If one node drops out of the network, due to hardware failure or any other reason, its neighbors simply find another route. Capacity can be increased by simply adding more nodes.

5.1 A Basic HSMM Radio Station

For the sake of simplicity, we'll discuss an HSMM setup with two stations operating in a host/client configuration. See **Fig 17**.

At the time of this writing, the most popular wireless router for HSMM applications is the Linksys model WRT54GL. It is a combination unit consisting of a wireless access point (AP) or hub coupled with a router. As with other routers, your host PC or laptop connects directly to it using a standard Ethernet cable. If the PC is also connected to the Internet, then it may also perform the function of a *gateway*.

The WRT54GL is a *Linux*-based model that supports firmware upgrades. HSMM-active amateurs have been creating their own WRT54GL firmware to support special applications (such as mesh networking). They are effectively "modifying" the WRT54GL in ways Linksys could not have imagined! You do not have the use a WRT54GL to explore HSMM, however. Any wireless router will do. The advantage of the WRT54GL is only that it is widely available and easily modifiable.

The first step in configuring your router for HSMM is to disconnect both flexible antennas that came with the unit and replace them with a high-gain directional antenna system. To connect a gain antenna, you are going to be become familiar with RP (reverse polarity) connectors. These connectors appear to be male connectors on the outside, but they have a socket rather than a pin for the center conductor. RP connectors are used by the manufacturers to discourage Part 15 owners from using the equipment in ways for which it was not WiFi certified.

As licensed radio amateurs, we can modify the system to accomplish our specific requirements within the amateur bands. To connect coaxial cable for an external antenna, you can use an adapter or short jumper cable assembly with appropriate connectors. The adapter shown in **Fig 18** has an RP connector on one end and a standard SMA connector on the other.

Fig 18 — RP connector adapters.

There are often two antenna ports on wireless devices. These are used for *receive space diversity*. The wireless device will normally automatically select whichever antenna is receiving the best signal at any specific moment. Which do you connect to?

The transmitted signal from the wireless router always goes out the same antenna port. It does not switch. In other words, most wireless devices use space diversity on the receive side and not on transmit. Some access points/wireless routers will allow you to manually

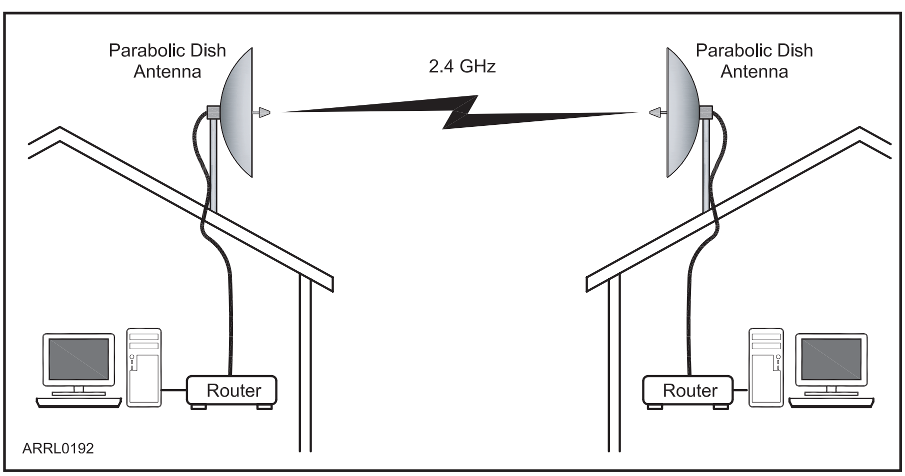


Fig 17 — This is a simplified diagram of a typical HSMM link between two stations. Note that high gain parabolic dish antennas are being used in this example.

(via software) select the antenna port that is used for transmission. When there is no obvious choice, you will need to find some means of detecting which antenna is the transmit antenna port with RF output power present. That is the port to connect to your gain antenna.

Be sure to use low-loss coaxial cable, and keep the run as short as possible. Coaxial cable loss at these frequencies can be quite high, so you are often better off mounting the router directly at the antenna, connecting to it through a very short length of cable. From there you can run the Ethernet cable back to your computer.

SOFTWARE CONFIGURATION

You will need to use either the network configuration software from the router manufacturer or the configuration tool that is part of your operating system. Some routers offer user-friendly access through Internet browsers, which is particularly convenient. Regardless of the approach, the goal is to access and change your router's settings.

SSID: The AP/wireless router host software is provided with an *SSID* (Service Set Identifier) that many Part 15 stations turned off for somewhat higher security. But radio amateurs should leave it ON. Enter your call sign as the SSID and use it for the station identification. It constantly broadcasts your call thus, providing automatic and constant station identification. There are 32 characters available for use in this field so more information such as your group's name can be entered too, including spaces and punctuation. If the router asks if you want to enable the broadcast, click YES. (Note that SSID in the HSMM world has a somewhat different meaning than SSID in the packet radio community. Among packet users, SSID is defined as Secondary Station Identifier.)

ESSID: Some manufacturers use this term in place of SSID to put emphasis on the fact that the SSID is the name for your *network,* not for a specific wireless AP/router.

Access Point Name: When this field is made available (by default is it blank) it is for you to enter a description. This may be handy if you have deployed more than one AP in your network all with your call sign as the SSID. It would allow you to tell them apart. Otherwise, just leave it blank.

Channel: To avoid interfering with other services as much as possible, we need to also look at channel selection. Most HSMM activity is concentrated on 2.4 GHz, using channels 1 through 6, which are within the 2.4 GHz ham allocation.

It is probably best to avoid channel 6. It is the most common manufactures default channel setting and 80% or more of your neighbors will be using it for their household wireless local area network (WLAN). Channel 1 is used by most of the remaining manufacturers as their default channel, so avoid that too. The result is most radio amateurs use channels 3 or 4 depending whether there is a WISP (Wireless Internet Service Provider) operating in their area. Often a WISP will use one of these intermediate channels with a highly directive antenna for back-haul or other purposes. If so, you may wish to coordinate with the WISP and arrange to use some other channel rather than the one specifically used by the WISP. It is not a perfect solution because of all the overlap, but it is a good faith effort to keep most of your stronger signal out of anyone's home, business or governmental WLAN traffic.

WEP: This stands for Wired Equivalent Privacy. In spite of all the horror stories you may have read in the press, this encryption method provides more than adequate means to economically achieve authentication and thus keep the vast majority of free-loaders off your network. If you live in the country you may not need to enable this capability. In an urban environment, it is probably a good thing to do so that you need not constantly monitor every bit of traffic coming over the network to ensure that it originates from an Amateur Radio station. Mixing traffic with another service that shares the same frequency band is not a generally accepted practice except in times of emergency. Therefore, it is often necessary for HSMM radio stations to encrypt their transmissions. This is not to obscure the meaning of the transmission or hide the information. In amateur HSMM applications, the purpose of using WEP is only to block access by non-hams. Amateur HSMM networks openly publish their encryption keys for other hams to use.

Most wireless routers will allow for the use of multiple WEP keys, typically up to four. This will allow you to configure the device so that different client stations have different access authority. For example, club members may have one level of access, while a visiting radio amateur may be given a lesser access. Most HSMM radio groups have just one WEP key and everybody gets that one.

Remember that when it comes to the length of the WEP or other key used, our main purpose is to provide a simple and economical means of authentication already available on the wireless devices. In other words, it is to ensure that only Part 97 stations and not Part 15 stations auto-associate or auto-connect with our HSMM radio node. The shorter the WEP key, the better. This makes it easy to use and remember. During early HSMM radio experimenting around the year 2000, the shortest possible key (5 characters) was used: HSMM-

Authentication Type: Some routers will ask for the type of authentication you want to use such as *shared key*, *open system*, and *both*. Click on SHARED KEY because you will be sharing the WEP/WPA key with any and all radio amateurs who wish to access your HSMM radio node.

*DHCP***:** Some routers will ask if you want Dynamic Host Configuration Protocol enabled. This is the function that assigns IP addresses. Unless you have another source of the DHCP function on your network, you will want to ENABLE this function.

Antenna Selection: A few wireless routers with dual antennas will ask you select an antenna. The default is normally receive space diversity. Because we are going to connect an outside gain antenna, you want to make a selection. Otherwise, you will need to identify which antenna is the actual transmit antenna and connect the feed line to that port, as discussed previously.

Mac Address Filter: Some wireless routers will allow for this security measure, but it is troublesome to administer it, so it is recommended that you not bother enabling this function. Use WEP or some other method of encryption using the guidelines discussed previously.

Output Power: Some wireless routers will allow you to set this power level, often up to 100 mW. As with all other radio amateur operations use only the minimum power needed to accomplish your mission.

THE FAR END OF THE LINK

The computer at the client end of the HSMM link need only have a wireless networking adapter, not a separate router. These transceivers/wireless adapter cards usually come in three forms:

1) One form is called a PC card. Earlier these were called PCMCIA cards, but more recent terminology is to simply call them laptop PC cards.

2) Another type of transceiver/adapter comes with a USB interface, such as the one shown in **Fig 19**. This is often considered a superior interface for most HSMM stations. The reason for this has nothing to do with the quality of the transceiver, but rather the fragile nature of the tiny connectors found on PC cards. They are not really designed for frequent plugging and unplugging. Without extreme care, they can be easily torn out.

3) Linksys and other manufactures also produce similar cards for the

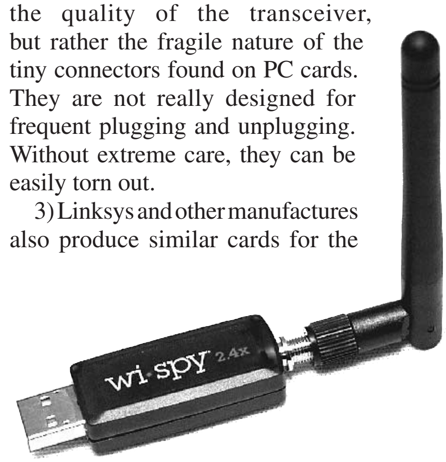


Fig 19 — At the client end of the HSMM link, a simple wireless adapter will do. This USB model allows the user to remove the flexible antenna.

expansion slot on the rear of your desktop PC too.

Regardless of the device you choose, it must have an antenna that is removable or has an external antenna port of some type.

To connect to the AP/wireless router in an HSMM radio network, the wireless computer user(s) at the far end must exit ad-hoc mode and enter what is called the *infrastructure mode*, in their operating software. Infrastructure mode requires that you specify the radio network your computer station is intended to connect to (the host station's call sign), so set your computer to recognize the SSID you assigned to the AP/wireless router to which you wish to connect.

As described previously, you may need to use an adapter to connect the card or interface antenna connector to the coaxial cable running to the antenna. At the other end, most 2.4 GHz antennas come with a standard female N-series connector.

Team up with a nearby radio amateur to test. Do your initial testing in the same room together making sure the link-up is working. Then as you increase distances going toward your separate station locations, you can coordinate using a suitable local FM simplex frequency. You will increasingly need this communication to assist with directional antenna orientation as you get farther apart.

5.2 Running High Power

It is tempting for some radio amateurs to think that if they run higher power they will get better range out of their HSMM radio station. This is not always the case. There are many factors involved in range determination when operating at 2.4 or 5.8 GHz. The first and most significant of these is the lay of the land (topology) and path obstructions. Running additional power is unlikely to correct for either of these impediments.

Second, running higher power to improve signal link margins often requires that this be done at both ends of the link to obtain meaningful results.

Third, most RF amplifiers for use with 802.11 are of the BDA (bidirectional amplifier) type, such as the one shown in **Fig 20**. They amplify both the incoming signal and the outgoing signal, and to get maximum effectiveness out of such devices they must be mounted as close as possible to the antenna.

Fig 20 — A 2.4 GHz bidirectional amplifier.

Fourth, 802.11 signals from such inexpensive broadband devices often come with significant sidebands. This is not prime RF suitable for amplification. A tuned RF channel filter should be added to the system to reduce these sidebands and to avoid splatter.

Also, if your HSMM radio station is next door to an OSCAR satellite ground station or other licensed user of the band, you may need to take extra steps in order to avoid interfering with them, such as moving to channel 4 or even channel 5. In this case a tuned output filter may be necessary to avoid not only causing QRM, but also to prevent some of your now amplified sidebands from going outside the amateur band, which stops at 2450 MHz.

Do not use higher power as a substitute for higher antenna gain at both ends of the link. Add power only after all reasonable efforts have been taken to get the highest possible antenna system gain and directivity.

5.3 HSMM Antenna Options

There are a number of factors that determine the best antenna design for a specific HSMM radio application. Most commonly, HSMM stations use horizontal instead of vertical polarization because it seems to work better. In addition, most Part 15 stations are vertically polarized, so this is sometimes thought to provide another small barrier between the two different services sharing the band. With multipath propagation is it doubtful how much real isolation this polarization change actually provides.

More importantly, most HSMM radio stations use highly directional antennas (**Fig 21**) instead of omnidirectional antennas. Directional antennas provide significantly more gain and thus better signal-to-noise ratios, which in the case of 802.11 modulations means higher rate data throughput and greater range. Higher data throughput, in turn, translates into more multimedia radio capability. Highly directional antennas also help amateurs avoid interference from users in other directions.

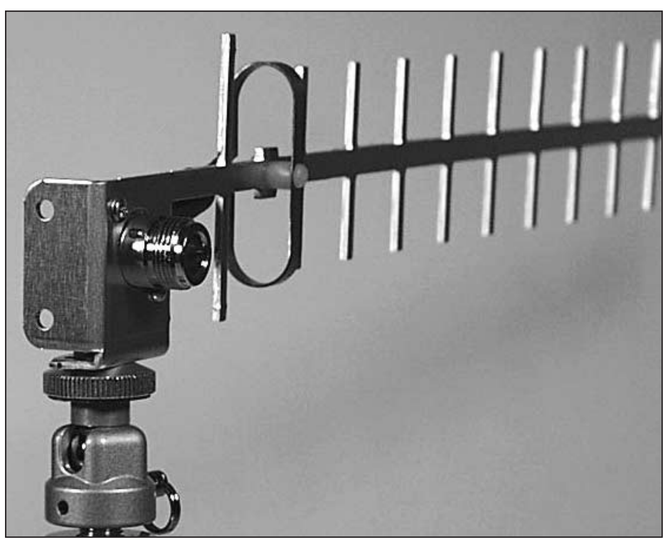


Fig 21 — Directional antennas, such as this MFJ 2.4 GHz Yagi, are best for HSMM links.

6 Automatic Link Establishment (ALE)

Automatic link establishment (*ALE*) is a "system" that incorporates digital techniques to establish a successful communications link. ALE is not a digital mode *per se*, but the primary software is sound card based.

To understand ALE, consider how HF propagation changes. A band can open over a particular path one hour, but close the next. A signal may be fully readable on one band, but inaudible on another. With that in mind, imagine two stations, one in New York City and one in Los Angeles, that wish to communicate on the HF bands (see **Fig 22**). Basic propagation guidelines give a general idea of which bands might be best, but operators at both ends of the path will have to experiment, calling on several bands until they find a frequency that supports a transcontinental path.

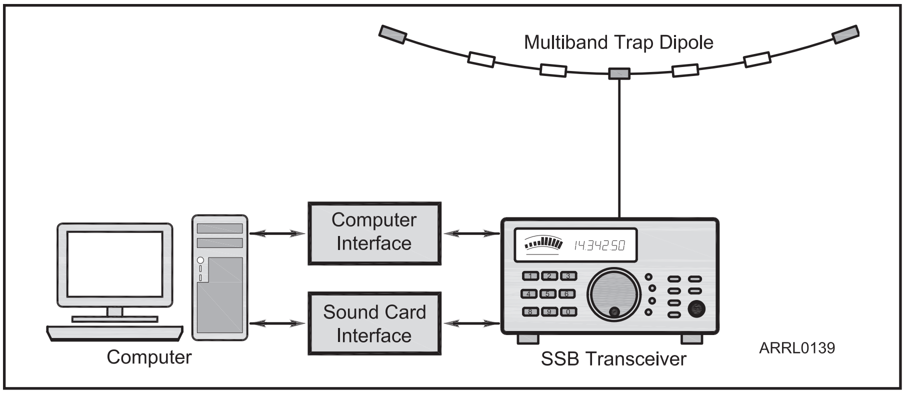


Fig 23 — An optimum ALE station has a computer-controlled transceiver and a multiband antenna system. In this example, the station is using a multiband trap dipole antenna. Note that a separate interface is needed between the computer and transceiver for frequency control.

ALE automates this process, allowing the stations under computer control to automatically call each other on different bands until contact is established. In the example, the operator in New York City initiates the ALE call by entering the call sign of the receiving station into his ALE software, along with some likely frequencies. The call sign becomes the *Selective Call*, or SELCAL. The ALE software switches the transceiver to the first frequency in the queue and transmits the chosen SELCAL. If there is no response, the software will step the transceiver to another frequency and try again.

At the same time, the receiving station in Los Angeles has been automatically scanning specific ALE frequencies, listening for its call sign. When it finally decodes something that resembles its call sign, it stops scanning and listens since the ALE data burst contains several repetitions of the call. When it finally decodes its full call sign, the Los Angeles station transmits a "handshake" signal to the New York City station to confirm that a link has been established. All scanning stops and the ALE software sounds an alarm to call the operators to their stations.

ALE is a good tool for HF net operation in both routine and emergency applications where many stations may be standing by for calls. Think of ALE as being analogous to a VHF FM scanning transceiver with a digitally coded squelch that automatically scans several repeaters. Unless it hears the correct code, the radio remains silent. With amateur ALE you can call individual stations or entire groups of stations.

In addition to its application as a selective calling system for HF voice, ALE can also be used as a pure digital mode for exchanging text and images.

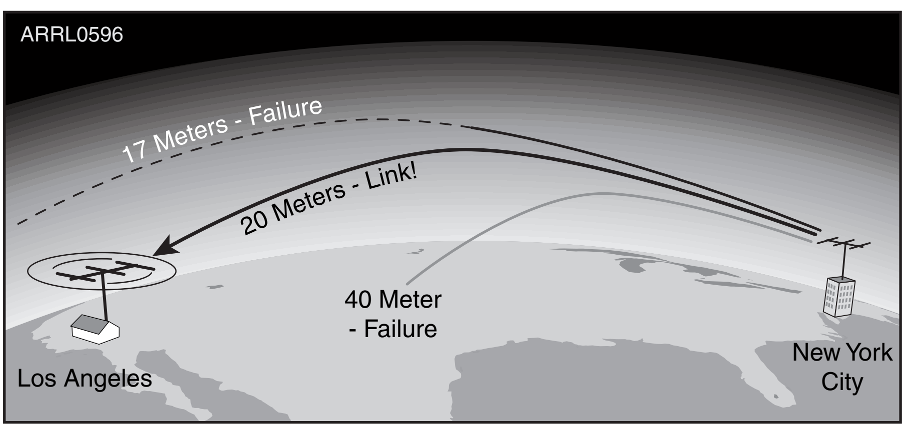


Fig 22 — The ALE calling station typically scans through a list of frequencies, calling on each one until a path to the receiving station is found.

AMATEUR ALE SOFTWARE

ALE has been used by the military and government for years, and hams have generally adopted the government ALE standards: FED-STD-1045 or MIL-STD 188-141. In 2001, ALE captured the attention of the Amateur Radio world with the release of *PCALE* software by Charles Brain, G4GOU. *PCALE* not only controls your transceiver, it uses your computer sound card to decode ALE signals and generate them for transmission. You'll find *PCALE* software on the Web at **www.hflink.com**. You must join the free HF Link group on Yahoo at **tech.groups.yahoo.com/group/hflink** before you can download the software.

From the standpoint of assembling a station for ALE, it works best when two conditions are satisfied: 1) you have a multiband HF antenna system; the more bands the better; and 2) your HF transceiver can be controlled by computer.

Although you can use ALE on just one frequency, the real power of ALE comes into play when your station is able to operate on multiple bands under computer control. A typical setup is shown in **Fig 23** and includes a radio control interface as well as a sound card interface. With *PCALE* controlling your radio, it will automatically change frequencies, monitoring as many bands and frequencies as you have programmed into the software. A list of ALE channels and information on ALE nets is available from **www.hflink.com**. Note that all ALE transmissions use USB.

7 D-STAR

The D-STAR digital protocol began in 2001 as a project funded by the Japanese Ministry of Posts and Telecommunications to investigate digital technologies for Amateur Radio. The research committee included representatives of the Japanese Amateur Radio manufacturers and the Japan Amateur Radio League (JARL). JARL is the publisher of the D-STAR protocol.

All radio manufacturers are free to develop D-STAR equipment, but at the time of this writing ICOM is the only company that has done so for the US amateur market. Although ICOM may develop D-STAR protocol enhancements peculiar to the function of its hardware, ICOM does not "own" the original D-STAR protocol.

7.1 What is D-STAR?

The primary application of D-STAR is digital voice, but the system is capable of handling any sort of data — including text, voice or images.

As shown in **Fig 24**, a D-STAR network can take several forms. D-STAR compatible transceivers can communicate directly (simplex), or through a D-STAR repeater for wide coverage. D-STAR signals cannot be repeated through traditional analog repeaters without modification.

The D-STAR system carries digitized voice and digital data, but does the job in two different ways. There is a combined voice-and-data mode (DV) and a high-speed data-only stream (DD). From the perspective of the D-STAR user, data and voice are carried at different rates and managed in different ways, but over the air they are transported as bit streams.

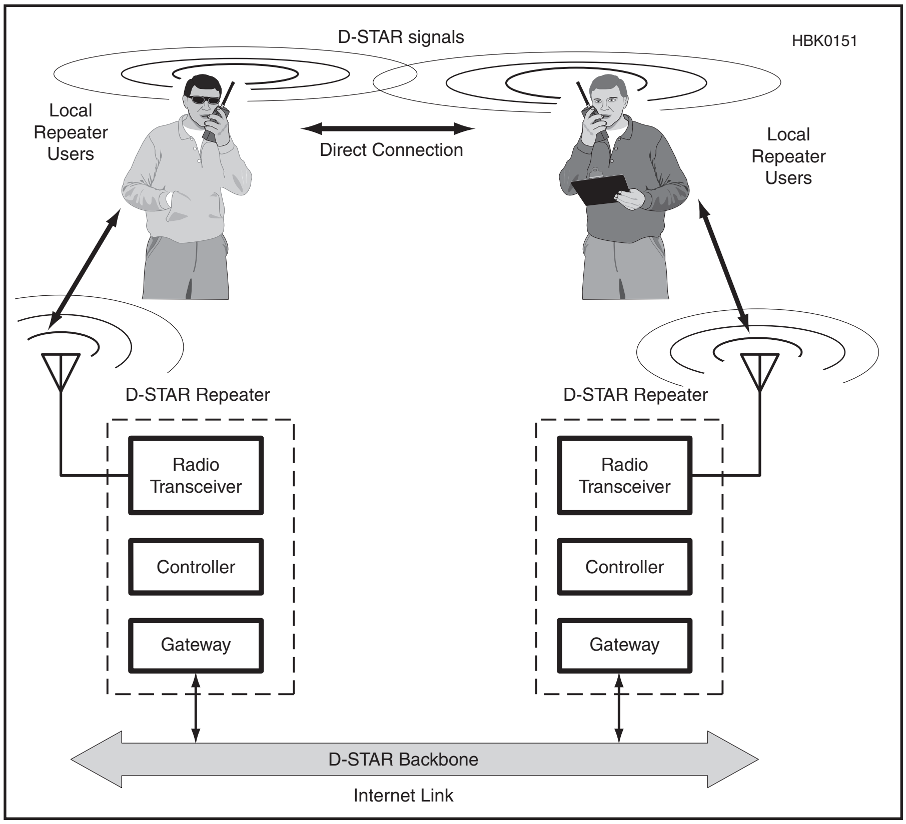


Fig 24 — A D-STAR network can take several forms. D-STAR compatible transceivers can communicate directly (simplex) or through a D-STAR repeater for wide coverage.

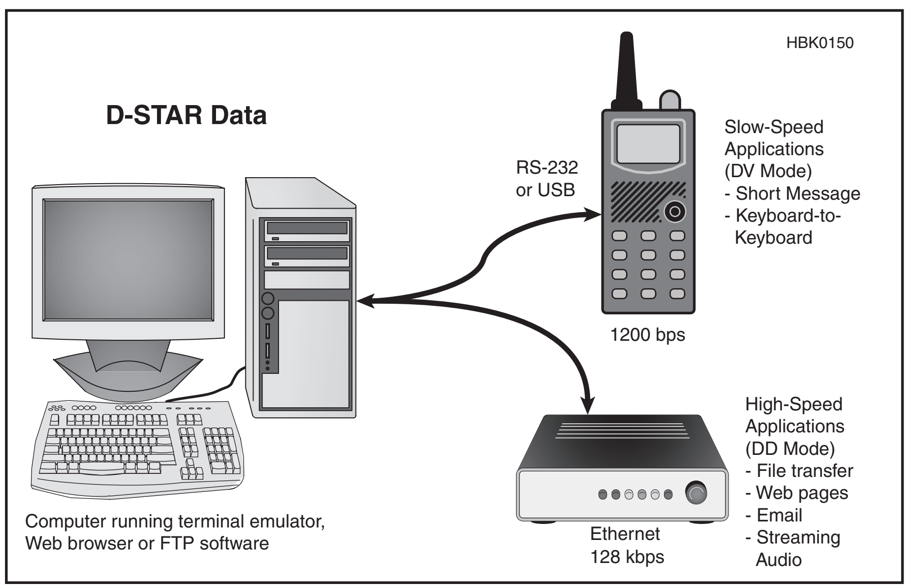


Fig 25 — Radios that support DV voice and data present an RS-232 or USB 2.0 interface to the user.

7.2 Digital Voice and Low-Speed Data (DV)

The D-STAR codec digitizes analog voice by using the AMBE 2020 codec. AMBE stands for Advanced Multiple Band Encoding and 2020 designates the particular variation used by D-STAR.

AMBE can digitize voice at several different rates. The D-STAR system uses a 2.4 kbit/s rate that offers a good compromise between intelligibility and the speed at which data must be transmitted via the radio link. In addition, AMBE adds information to the voice data that allows the receiving codec to correct errors introduced during transmission. The net result is that the digitized voice stream carries data at a rate of 3.6 kbit/s.

Interleaved with the digitized voice information, D-STAR's DV mode also carries 8-bit digital data at 1200 bit/s. This data is used for synchronization in the D-STAR protocol with the remaining bandwidth available for use by the radio manufacturer. ICOM uses the remaining bandwidth to carry repeats of the D-STAR RF header, the 20 character front-panel message, and serial data as described below.

Radios that support DV voice and data provide an RS-232 or USB 2.0 interface to the user as shown in **Fig 25**. (The RS-232 interface is restricted to RxD, TxD and ground — "three-wire" connection.) Any computer terminal or program that can exchange data over those types of interfaces can use D-STAR's DV mode capabilities as a "radio cable."

Because D-STAR's DV mode handles the data stream in an unmodified "raw" format, it is up to the equipment or programs that are exchanging data to manage its flow. ICOM requires the sender and receiver to perform flow control by using special data characters. This is called *software flow control or xon/xoff flow control*.

7.3 High-Speed Data (DD)

D-STAR's high-speed data mode is called *D-STAR DD*. This mode does not digitize the analog voice signal. The data-only packets sent over the RF link at a raw data rate of 128 kbit/s half-duplex, but since that includes the packet header and the delay between packets, the *net data rate* is somewhat lower. As with the DV mode, data is transmitted without modification so flow control is left to the applications on each end. Radios supporting DD mode communications may also support DV mode.

Users connect to a radio supporting DD mode with an Ethernet interface via the usual RJ-45 modular jack found on computer networking equipment. The DD mode interface looks to computer equipment just like a customary IP network connection. Specifically, the DD mode interface is an Ethernet "bridge." This allows Web browsers and other Internet software to run normally, as if they were connected to standard computer network.

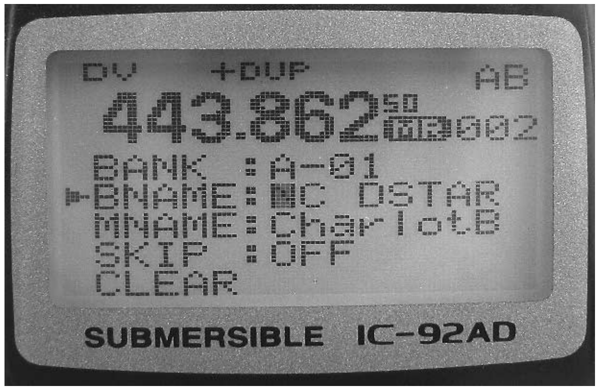


Fig 26 — The ICOM IC-92AD is a 2 meter/70-cm handheld with D-STAR functionality.

The net data rate of DD mode is comparable to or better than a high-speed dial-up Internet connection. Any streaming media mode that will run over dial-up Internet will likely perform well over D-STAR.

With its high signaling rate and 130-kHz bandwidth, FCC regulations restrict D-STAR DD operating to the 902 MHz and higher bands.

7.4 D-STAR Radios

When this book went to press, ICOM was the only manufacturer offering D-STAR compatible radios and repeaters. So, by default, the focus of any discussion of D-STAR hardware centers on ICOM gear.

ICOM D-STAR radios are available as mobile/base transceivers and also as handheld transceivers. With the exception of the ID-1 base/mobile transceiver, which operates at 1.2 GHz, ICOM D-STAR transceivers are designed to operate at 2 m, 70 cm or both. ICOM D-STAR transceivers also support both digital and analog operation, which means that they can also be used for traditional analog FM communication as needed. An example is shown in **Fig 26**.

To determine the most appropriate D-STAR radio for an application, the first step is to understand the requirements:

- Is DD mode operation required (high-speed data)? If so, the only radio supporting DD mode is the ID-1.
- What bands are required? That will depend on activity in your area. If data is to be transmitted while in motion, higher frequencies will result in fewer transmission errors, improving the net data exchange rate. (Note: DD operation is only permitted on 902 MHz and higher.)
- Is high-power required? Using higher power base/mobile radios will result in stronger signal strengths and fewer data transmission errors.
- If data is to be transmitted, what data interface does the computer have?
- Error correction for low-speed data (DV mode) is the responsibility of the data communications programs used to exchange data. Make sure to carefully evaluate the software you intend to use.

More information about D-STAR may be found in the **Digital Modes** and **Repeaters** chapters.

8 APCO-25

Unlike D-STAR, which is a digital standard devised by and for Amateur Radio, APCO-25 was developed specifically for local, state and federal public safety communications. "APCO" is the Associated Public-Safety Communications Officials, originally an association of police communication technicians, but now a private organization. APCO has a technical standards group responsible for planning the future needs of police (and more recently public safety) users. It was through this group that a standard for advanced narrow-band digital communications (voice or data) was developed. This standard is known as APCO Project 25, APCO-25, or simply P25.

The overall purpose of the APCO-25 standard is to make it possible for governments to shift from analog to digital communications with the least difficulty possible. This means placing a great deal of emphasis on *backward compatibility* (P25 radios include analog operation and newer P25 technology doesn't render older technology instantly obsolete) and *interoperability* (the ability for all P25 radios to communicate with each other). In the public safety world, interoperability is a key selling point so that various services and agencies can coordinate efforts.

8.1 APCO-25 and Amateur Radio

When this section was written, no one was making APCO-25 transceivers specifically for the Amateur Radio market, but that hasn't stopped some hams from exploring this mode. Because APCO-25 is an open, published standard, it *is* legal for Amateur Radio, but the trick is finding the means to adapt commercial APCO-25 gear to ham purposes.

The present-day Amateur Radio APCO-25 world looks a lot like the analog FM community in the early 1970s. Back then, none of the Amateur Radio manufacturers were making FM transceivers or repeaters. Hams were forced to modify existing commercial FM gear, which typically consisted of transceivers that had seen duty in police cruisers, taxi cabs and so on. Many of the FM "gurus" in those days were individuals who were employed by two-way radio service shops. These hams had easy access not only to test equipment, but also to the knowledge of how to modify commercial transceivers for ham applications. They built the first Amateur Radio FM repeaters by repurposing commercial two-way radio transmitters and receivers.

Amateur Radio APCO-25 enthusiasts today are treading the same path taken by analog

FM pioneers more than 30 years ago. They are modifying commercial APCO-25 equipment for Amateur Radio and setting up APCO-25 repeater systems. Thanks to online sites such as eBay, it is relatively easy to track down surplus APCO-25 transceivers. Manufactured in both handheld and mobile configurations, these rigs are available for either VHF or UHF.

Modifying commercial APCO-25 radios has become a software hacking game. Since most functions of these radios are software defined, you can change their operating characteristics (including frequencies) with a computer and a compatible interface.

The first step is to obtain the programming software, which can be different for every brand. Some manufacturers provide programming software if you purchase the radio as new equipment. Others are highly restrictive and will not provide their software under any conditions. Motorola, a popular brand among amateur P25 users, uses different programs for every transceiver model. The programming software must be purchased from them at a cost of $250 to $300. To reprogram the Motorola transceivers (as well as many other brands), you may need a hardware device that is sometimes referred to as a Radio Interface Box, or RIB.

The downside of using surplus commercial equipment is that it can be expensive. At the time of this writing, used handheld APCO-25 transceivers were selling for as much as $700 at Internet auction sites. Modified APCO-25 repeaters can cost several thousand dollars.

The cost hurdle hasn't deterred hams from setting up APCO-25 networks. There are more than a dozen amateur APCO-25 repeater systems in operation throughout the United States. Many of these repeaters operate in mixed mode — analog and digital. Others are digital only. The ability to operate in mixed modes is one of the strengths of Amateur Radio APCO-25. An APCO-25 repeater, for instance, can support digital voice and data with APCO-25 transceivers, but can still relay analog FM traffic.

More ham-related programming information is available at the following Web sites, although it applies specifically to Motorola transceivers: **www.batlabs.com/newbie.html** and **www.batlabs.com/flash.html.**

8.2 The APCO-25 Standard

APCO-25 is comprised of a "Suite of Standards" that specifies eight open interfaces between the various components of a land mobile radio system.

• *Common Air Interface* (CAI) standard specifies the type and content of signals transmitted by compliant radios. One radio using CAI should be able to communicate with any other CAI radio, regardless of manufacturer

• *Subscriber Data Peripheral Interface* standard specifies the port through which mobiles and portables can connect to laptops or data networks

• *Fixed Station Interface* standard specifies a set of mandatory messages supporting digital voice, data, encryption and telephone interconnect necessary for communication between a Fixed Station and P25 RF Subsystem

• *Console Subsystem Interface* standard specifies the basic messaging to interface a console subsystem to a P25 RF Subsystem

• *Network Management Interface* standard specifies a single network management scheme which will allow all network elements of the RF subsystem to be managed

• *Data Network Interface* standard specifies the RF Subsystem's connections to computers, data networks, or external data sources

• *Telephone Interconnect Interface* standard specifies the interface to Public Switched Telephone Network (PSTN) supporting both analog and ISDN telephone interfaces.

• *Inter RF Subsystem Interface* (ISSI) standard specifies the interface between RF subsystems which will allow them to be connected into wide area networks.

You'll find more details about the APCO-25 standard on the Web at **www.apcointl.org/frequency/project25/index.html**.

8.3 APCO-25 "Phases"

The APCO-25 rollout was planned in "phases." Phase 1 radio systems operate in 12.5 kHz analog, digital or mixed mode. Phase 1 radios use continuous 4-level FM (C4FM) modulation for digital transmissions at 4800 baud and 2 bits per symbol, which yields 9600 bits per second total throughput. It is interesting to note that receivers designed for the C4FM standard can also demodulate the compatible quadrature phase shift keying (CQPSK) standard. The parameters of the CQPSK signal were chosen to yield the same signal deviation at symbol time as C4FM while using only 6.25 kHz of bandwidth. This is to pave the way for Phase 2, which is under development. Phase 1 is the current phase in use at the time this book was written and is likely to be in force for a number of years to come.

In a typical Phase 1 radio, the analog signal from the microphone is compressed and digitized by an Improved Multi-Band Excitation, or *IMBE*, vocoder. This is a proprietary device licensed by Digital Voice Systems Corporation. The IMBE vocoder converts the voice signal from the microphone into digital data at a rate of 4400 bit/s. An additional 2400 bit/s worth of signaling information is added, along with 2800 bit/s of forward error correction to protect the bits during transmission. The combined channel rate for IMBE in P25 radios is 9600 bit/s.

P25 radios are able to operate in analog mode with older analog radios and in digital mode with other P25 radios. If an agency wants to mix old analog radios with P25 radios, the system must use a control channel that both types of radios can understand. That means a trunked radio system.

9 HF Digital Voice

While D-STAR is the dominant digital voice/data system on the VHF+ bands, three very different systems can be found in operation on the HF bands. One uses dedicated hardware while the other two rely on sound cards and software. All require HF SSB transceivers, although they could just as easily be used on VHF SSB (or even FM) as well.

9.1 AOR and AMBE

The AOR Corporation was the first ham manufacturer to arrive on the scene with an HF

Fig 27 — The AOR ARD-9800 digital voice modem.

digital voice and data "modem" in 2004. The fundamental design is based on a Vocoder protocol created by Charles Brain, G4GUO. His protocol involves the use of Advanced Multi-Band Excitation, better known as *AMBE,* a proprietary speech coding standard developed by Digital Voice Systems. Brain's protocol operates at 2400 bit/s, with Forward Error Correction added to effectively produce a 3600 bit/s data stream. This data stream is then transmitted on 36 carriers, spaced 62.5 Hz apart, at 2 bits/symbol, 50 symbols/s using QPSK. This gives the protocol a total RF bandwidth of approximately 2250 Hz (compared to 2700-3000 Hz for an analog single sideband transmission).

The AOR units, such as the ARD9800 (**Fig 27**), are designed to be as "plug and play" as possible. You simply plug your microphone into the front-panel jack and then plug the modem into the microphone input of your transceiver. A front panel switch selects digital encoding or analog transmission, so the unit can be kept in the line when not used for digital conversations. On the receive side, the modem automatically detects the synch signal of an AMBE transmission and switches to digital mode automatically. In addition to voice, the AOR modems can send digital data (typically images).

AOR ON THE AIR

AOR digital voice is clear and quiet, an unusual thing to hear on an HF frequency. Tune around 14.236 MHz and you're likely to hear AOR AMBE signals. They sound like rough hissing noises.

During *QST* Product Review testing in 2004, the ARRL Lab discovered that decoding was solid down to about 10 dB S/N. Digital voice is an all-or-nothing proposition, however, and below that level the signals begin to break up. The result is absolute silence during those periods. Interference to the received signal produced a similar outcome.

The common approach on the air is to begin the conversation in analog SSB, then switch to digital. Each transmission starts with a one-second synch burst after the push-to-talk button is pressed. If the other station misses the sync signal, the audio doesn't decode and the transmission sounds like analog white noise. This means that for successful communication you must be on the correct frequency and ready to receive at the beginning of the transmission. However, if there is fading or interference during the synch transmission, you may be unable to decode the signal that follows.

A few additional steps will improve odds of success with AOR modems.

- Make sure that both stations are on exactly the same frequency, within about 100 Hz.
- Set IF receive filters to 3 kHz or wider.

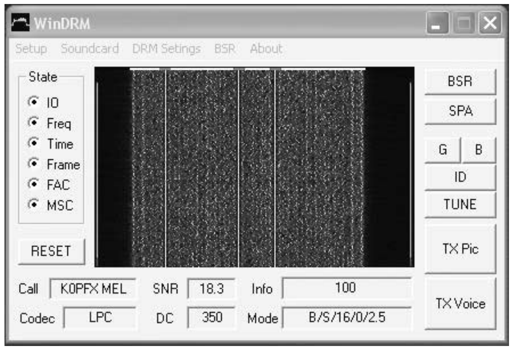


Fig 28 — *WinDRM* digital voice/image software.

- Don't overdrive the modem audio input.
- Turn off transceiver speech compression.
- Don't overdrive the radio. If the ALC meter shows any activity, turn down the output from the modem.

Transceiver duty cycle will be substantially higher when you're operating digital voice compared to normal SSB. To avoid damaging your radio, it is a good idea to reduce your output by 25 to 50%.

9.2 WinDRM

WinDRM began with *Dream*, a piece of open-source software developed by Volker Fisher and Alexander Kurpiers at the Darmstadt University of Technology in Germany. It was designed to decode Digital Radio Mondiale (DRM), a relatively new digital shortwave broadcast format. On the air, DRM presents as a wide, roaring signal. A commercial DRM signal is capable of carrying high-quality audio along with text and occasional images.

DRM uses Coded Orthogonal Frequency Division Multiplexing (COFDM) with Quadrature Amplitude Modulation (QAM). COFDM uses a number of parallel subcarriers to carry all the information, which makes it a reasonably robust mode for HF use.

Not long after *Dream* debuted, Francesco Lanza, HB9TLK, began adapting it for Amateur Radio. He redesigned *Dream* to support amateur DRM transmission and reception within a 2.5-kHz SSB transceiver bandwidth. That meant sacrificing some audio quality, along with the ability to simultaneously send images.

As the name implies, *WinDRM* is a *Windows* application (**Fig 28**). It is designed to use a computer sound card or on-board sound chipset to send and receive amateur DRM. Activity is primarily on HF. In addition to the software, you need a radio, computer and one of the sound-card interfaces discussed earlier in this chapter. Start by downloading and installing the *WinDRM* software at **n1su.com/windrm/download.html**. On this same Web page you will find documentation that explains the installation and operation.

To receive amateur DRM, all you need is a cable between the audio output of your radio and the LINE INPUT of your sound card. You may need to get into your sound card control software and boost the LINE INPUT gain. Using the *Windows* audio mixer, which you can access within *WinDRM*, bring up the **Recording Control** panel and make sure LINE is selected and that the "slider" control is up. If the *WinDRM* waterfall display is too bright, reduce the gain control.

Transmitting with *WinDRM* is somewhat more complicated in terms of your station setup, especially if you have only one sound card. The audio output from your sound card must be applied to your headset or PC speakers for receiving, but the *same* output must also feed your transceiver for transmitting (see **Fig 29**) so a switch for the audio output is needed. The easier, more elegant approach is to use two separate sound cards as shown in **Fig 30**. The second sound card doesn't need to be high-end; you can use an inexpensive USB sound card for this application.

As with the AOR modems, be careful not to overdrive your transceiver. If you see the ALC meter indicating excessive drive, reduce the sound card audio output.

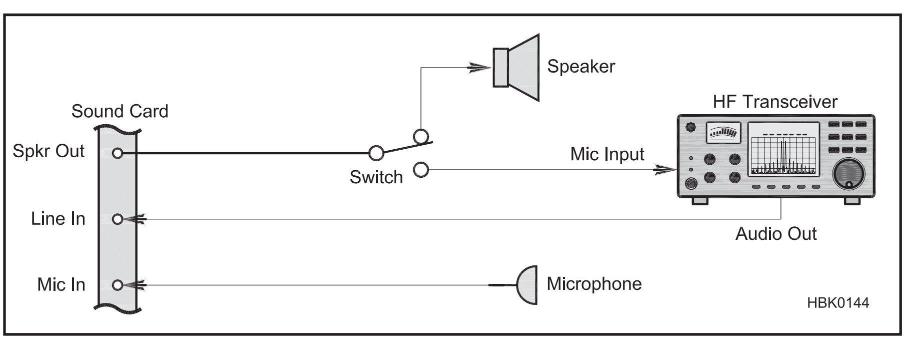


Fig 29 — The single sound card approach to using *WinDRM* or *FDMDV* requires the means to switch the audio stream when switching from transmit to receive.

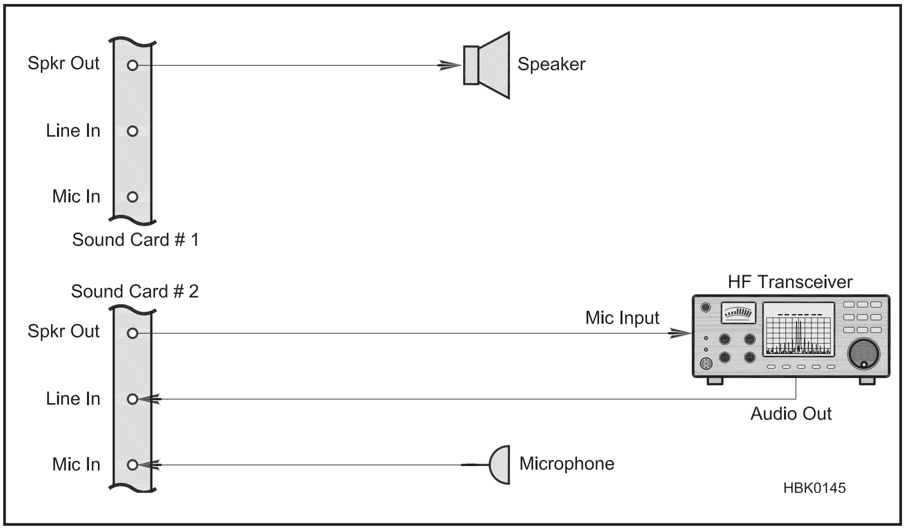


Fig 30 — The most elegant, easy-to-operate solution for *WinDRM* and *FDMDV* is to use two sound cards — one to process the received signal and the other to generate the transmit signal.

9.3 FDMDV: Frequency Division Multiplex Digital Voice

Like *WinDRM*, *FDMDV* is a software approach to digital voice that uses the sound card for processing the received signal and generating the transmit signal. The *FDMDV* software is available at **www.n1su.com/fdmdv**, along with the documentation. Station setup and hardware requirements are identical to *WinDRM*.

The advantages of FDMDV compared to WinDRM are twofold: 1) FDMDV can decode signals at substantially weaker levels and 2) FDMDV occupies an effective bandwidth of only 1100 Hz.

A popular frequency for HF digital voice is 14.236 MHz, USB. Information about digital voice nets and activity on other bands is available from **www.n1su.com** and other Web sites.

10 EchoLink, IRLP and WIRES-II

Worldwide communication on HF requires reliable propagation and more substantial radio and antenna requirements than a basic VHF FM setup. By using Internet links instead of HF radio links, a ham with a modest radio (even a handheld transceiver) can reliably communicate with stations hundreds or even thousands of miles away at any time of the day or night.

The most common form of amateur Internet linking involves the exchange of audio using *VoIP* — Voice over Internet Protocol — technology. This is the same technology used by Internet telephone services, and by online voice "chat" applications such as *Skype*, *TeamSpeak* and others. Three versions of Amateur Radio voice linking have become popular in the US: EchoLink, IRLP and WIRES-II.

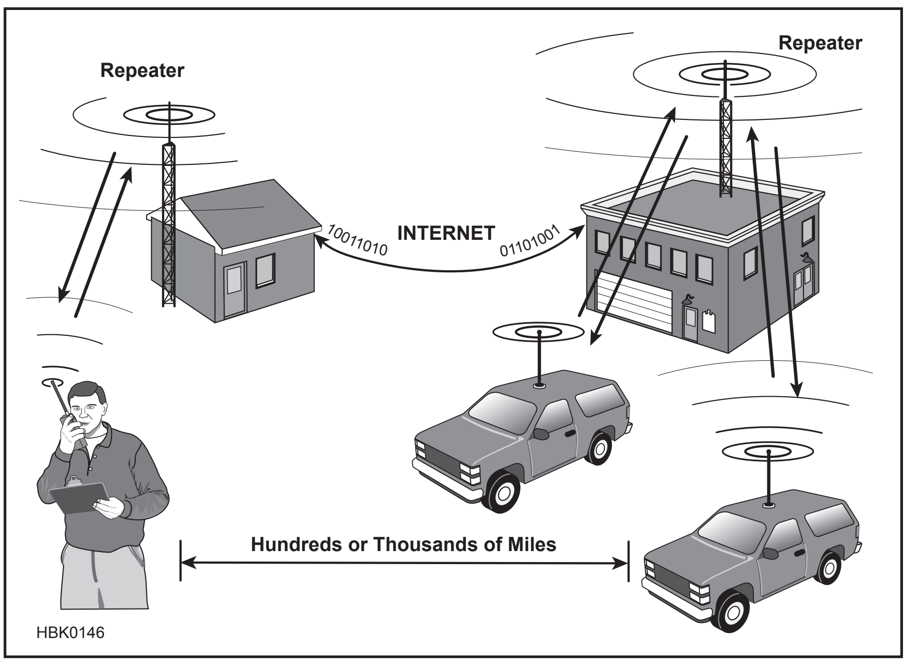


Fig 31 — The Internet VoIP link allows repeater users to speak with other amateurs who may be transmitting through other repeaters.

10.1 EchoLink

EchoLink software, developed by Jonathan Taylor, K1RFD, is designed for *Windows* PCs, but there is a Mac version as well. With the software installed on a sound-card-equipped computer, any ham can create an EchoLink *node* that others can access by radio. Hams can also join the network directly without using radios. They simply plug microphone/headsets into their computers.

Each EchoLink node is assigned a number that can be up to six digits in length. RF users access the EchoLink network by first sending the DTMF code required to activate the link on a repeater or simplex node. Of course, this requires a transceiver equipped with a DTMF keypad. Once the link is up, they use the keypad again to transmit the number of the EchoLink repeater, link or user they wish to contact.

Hams who access the network directly (without a radio) select the person or system they wish to contact by simply clicking on its name in the software listing. The EchoLink network maintains continuously updated databases on several servers that indicate node and user locations and operational status. The network servers also support *conferencing* where many users can "meet" and speak together, either by direct access or via radio.

EchoLink supports communication among three different source groups:

• *Repeaters*: This is typically a VHF/UHF FM repeater with a computer at the site and a

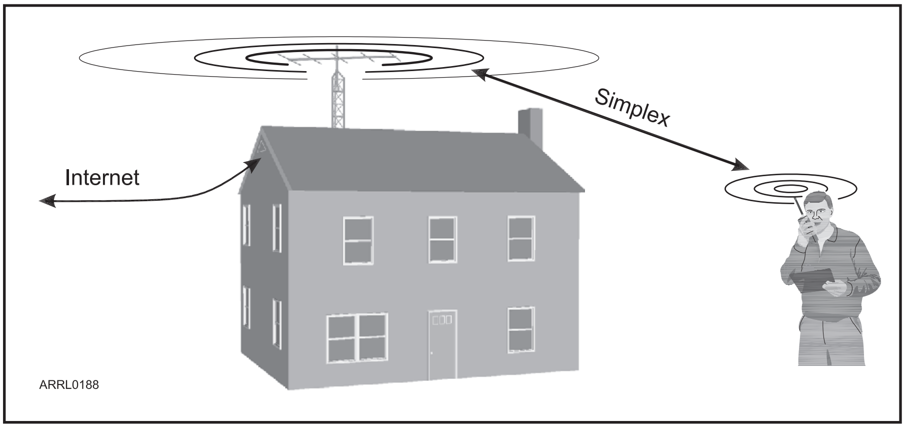


Fig 32 — A typical Echolink simplex node.

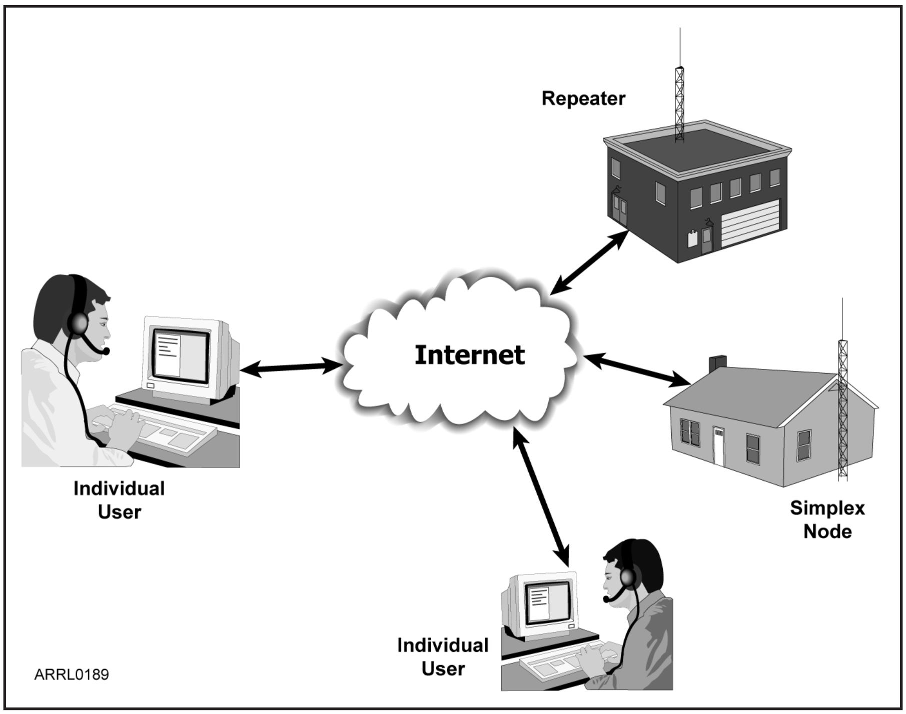


Fig 33 — For some EchoLink users, there is no radio involved (at least at their end of the conversation).

VoIP connection to the Internet via EchoLink. When the Echolink function is switched on, audio from the repeater is available on the EchoLink network, and vice versa. The Internet VoIP link allows repeater users to speak with other amateurs who may be transmitting through other repeaters, or who may be linking directly (see **Fig 31**).

• *Links*: Similar to repeater nodes, but operating on simplex frequencies without full repeater functionality (see **Fig 32**). These are typically FM transceivers connected to computers. Depending on the station setup, their coverage can be confined to a neighborhood, or it may encompass an entire county.

• *Users*: This is strictly a computer-to-computer VoIP connection (see **Fig 33**) with no RF involved. In other words, the operator is sitting in front of his computer wearing a microphone/headset.

All of the interfacing on the "RF side" of EchoLink is handled by connections to the computer sound card and serial port with a sound-card interface used to switch the radio between transmit and receive — exactly the same setup used by sound card based digital communications such as PSK

ECHOLINK SECURITY

Before being granted access, every ham who establishes an EchoLink node, or who connects to the network directly, must provide positive proof of identity and license during the EchoLink registration process. Details are available from **www.echolink.org/authentication**. After having been validated, each EchoLink node or user must provide a password, along with his or her call sign, to log in. Each time a connection is made for a QSO, the EchoLink servers verify both the sender and the receiver before communication can begin. Hams who use their radios to connect to the EchoLink network through a repeater or a simplex node do *not* have to be validated or provide a password. This is only necessary for the repeater or link stations, or for those who connect directly.

It is possible to configure EchoLink to accept connections only from certain types of stations: repeaters, links, users or all three. You can also set up a list of any number of "banned" call signs, which will not be allowed access. In addition, you can block or accept connections according to their international call sign prefix, in order to comply with reciprocal control-operator privileges or third-party traffic restrictions.

In Sysop mode, by default, EchoLink announces each station by call sign when the station connects. (The user can program a voice or CW ID that is generated automatically when needed.) The EchoLink software automatically generates detailed logs and (optionally) digital recordings of all activity on the link.

Unlike software such as e-mail programs, file-sharing programs and Web browsers, EchoLink does not have any way to pass files or "attachments" that might harm your computer. There are no known cases of EchoLink accepting or spreading a computer virus. Of course, any PC connected to the Internet should always be protected by some sort of Internet security hardware or software.

10.2 IRLP

IRLP stands for Internet Radio Linking Project. Like EchoLink, IRLP uses the Internet to establish VoIP links. The difference is that IRLP only permits access via RF nodes at repeaters or on simplex frequencies (all IRLP nodes are interlinked via a central Internet server). You must use a radio to access the IRLP network.

A typical IRLP node consists of a transceiver that is connected to the Internet through dedicated IRLP hardware and software. This requires a *Linux* computer running IRLP software and an IRLP interface board, or a turnkey "embedded" IRLP unit that combines the *Linux* computer, IRLP software and interface in a single package (**Fig 34**). The IRLP software controls the VoIP audio stream using carrier operated squelch (COS) or continuous tone coded subaudible squelch signals (CTCSS) from the transceiver. When COS is present, the computer detects it through the IRLP interface board.

The operator connects to the IRLP network (at a repeater, for instance) using DTMF sig-

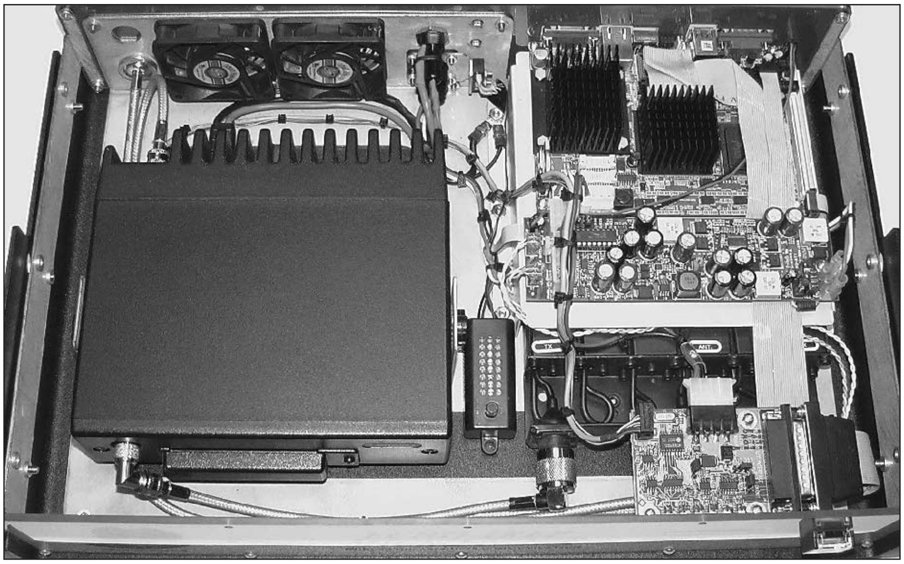


Fig 34 — The Micro-Node MN-3200 is an embedded IRLP node that combines the *Linux* computer, IRLP software and interface, along with a 70 cm repeater and duplexer in a single package.

nals sent over the radio. As with EchoLink, this means that the operator must have a radio with a DTMF keypad. The actual access code is determined by the owner of the node. The user sends the correct DTMF sequence to bring up the IRLP connection, then sends the sequence of the distant node he wishes to contact. That might be a repeater in another state or a distant simplex node. You can view a list of current IRLP nodes at **status.irlp.net/statuspage.html**.

Here is an example of a typical IRLP connection:

• Press the transceiver PTT and identify your station. For example, "This is WB8IMY, Steve in Wallingford, Connecticut, USA connecting to node 8880."

• Next, press *#8880* (node 8880 in this example) on the DTMF keypad, release PTT and wait for the node connect announcement. The announcement indicates that your audio is now reaching the distant node. You can call a specific person, or just ask if anyone is around to chat.

• When finished, key PTT , make an announcement such as "WB8IMY releasing node 8880" and then press *#73* on the DTMF keypad to disconnect. The IRLP node will announce the disconnect and the node may now connect to a different node, reflector, or it will stand by awaiting another user.

It is interesting to note that there is a project underway (known as *EchoIRLP*) to bridge the IRLP and EchoLink networks.

10.3 WIRES-II

WIRES-II — Wide-Coverage Internet Repeater Enhancement System — is a VoIP network created by Vertex Standard (Yaesu). It is functionally similar to IRLP (including management via a central Internet server), but nodes and repeaters are connected via a Vertex Standard HRI-100 interface and a computer running under the *Windows* operating system. Vertex Standard transceivers are *not* required for WIRES-II communication, but as with IRLP, all access must be via RF.

There are two WIRES-II operational configurations using home stations (simplex nodes), repeaters or a combination of the two:

SRG — Sister Radio Group. You operate within WIRES-II in a small (10 node maximum) network that is ideal for closed-group operations. Within the network, all nodes operate using the same repeater node list, so you can link only to stations within this 10 node network. Because there are only ten nodes maximum, access to any of these nodes is possible using a single DTMF tone when calling. At the beginning of each transmission, this single DTMF tone locks communication between the calling node and the called node, but local (non-linked) transmissions are also possible, simply by omitting the DTMF tone at the beginning of the transmission.

FRG — Friends Radio Group. You may call any repeater registered with the WIRES-II FRG server. In the case of FRG operation, a six-digit DTMF code is required for access, and once the link is established this code need not be sent again (this is called the LOCK mode), unless the operator wants the ability to make non-linked transmissions (UNLOCK mode), in which case the six-digit code must be sent at the beginning of each transmission (using the DTMF Autodial feature of the transceiver, for example). Group calling to preset 10-repeater B, C, and D lists is also possible.

SRG is similar in philosophy to the local FM voice repeater where you tend to talk to the same group on a regular basis. FRG is similar in concept to IRLP, linking together a worldwide group of repeaters and base stations.

11 Glossary of Digital Communications Terms

***AFSK* —** Audio frequency shift keying, a method of digital modulation in which audio tones of specific frequencies are used with an SSB voice transceiver.

***ALE* —** Automatic link establishment. A process in which stations under computer control automatically call each other on different bands until contact is established.

AMBE — Advanced multi-band excitation, a proprietary speech coding standard developed by Digital Voice Systems.

***AMTOR* —** Amateur teleprinting over radio, an amateur radioteletype transmission technique employing error correction as specified in several ITU-R Recommendations M.476-2 through M.476-4 and M.625.

APRS — Automatic Packet (or Position) Reporting System. A system of sending location and other data over packet radio to a common Web site for tracking and recording purposes.

***ARQ* —** Automatic Repeat reQuest, an error-sending station, after transmitting a data block, awaits a reply (ACK or NAK) to determine whether to repeat the last block or proceed to the next.

***ASCII* —** American National Standard Code for Information Interchange, a code consisting of seven information bits.

***AX.25* —** Amateur packet-radio link-layer protocol. Copies of protocol specification are available from ARRL HQ.

***Baud* —** A unit of signaling speed equal to the number of discrete conditions or events per second. (If the duration of a pulse is 20 ms, the signaling rate is 50 bauds or the reciprocal of 0.02, abbreviated Bd).

***Baudot code* —** A coded character set in which five bits represent one character. Used in the US to refer to ITA2.

***BER* —** Bit error rate.

***BERT* —** Bit-error-rate test.

Checksum — The output of an algorithm that allows the receiving system to detect errors in transmitted data.

***CLOVER* —** Trade name of digital communications system developed by Hal Communications.

***Collision* —** A condition that occurs when two or more transmissions occur at the same time and cause interference to the intended receivers.

***CRC* —** Cyclic redundancy check, a mathematical operation. The result of the CRC is sent with a transmission block. The receiving station uses the received CRC to check data integrity.

Data modes — Computer-to-computer communication, such as by **packet radio** or **radioteletype (RTTY)**, which can be used to transmit and receive computer characters, or digital information.

Digipeater — A station that relays digital data transmissions.

DRM — Digital Radio Mondiale, a digital modulation method used to transfer audio and data on HF bands.

EchoLink — A system of linking repeaters and computer-based users by using the Internet (also see **VoIP**).

***Eye pattern* —** An oscilloscope display in the shape of one or more eyes for observing the shape of a serial digital stream and any impairments.

***FEC* —** Forward error correction, an error-control technique in which the transmitted data is sufficiently redundant to permit the receiving station to correct some errors.

FSK — Frequency shift keying, a method of digital modulation in which individual bit values are represented by specific frequencies. If two frequencies are used, one is called *mark* and one *space*.

GPS — Global Positioning System

***G-TOR* —** A digital communications system developed by Kantronics.

***Hellschreiber* —** A facsimile system for transmitting text.

HSMM — High speed multimedia, a digital radio communication technique using spread spectrum modes primarily to simultaneously send and receive video, voice, text and data.

IEEE 802.11 — An IEEE standard for spread spectrum communication in the 2.4 GHz band at 1 Mbit/s and 2 Mbit/s data rates. 802.11 is also used as a general term for all spread spectrum devices operating under Part 15. (Also see **WiFi**.)

IEEE 802.11a — An IEEE standard for spread spectrum communication in the 5.8 GHz band at 6, 12, 16, 24, 36, 48, and 54 Mbit/s data rates.

IEEE 802.11b — An IEEE standard for spread spectrum communication in the 2.4 GHz band at 5.5 and 11 Mbit/s data rates in addition to being backward compatible with DSSS at 1 and 2 Mbit/s specified in 802.11.

IEEE 802.11g — An IEEE standard for spread spectrum communication in the 2.4 GHz band at 6, 12, 16, 24, 36, 48, and 54 Mbit/s data rates in addition to being backward compatible with DSSS at 1, 2, 5.5, and 11 Mbit/s specified in 802.11b.

IEEE 802.11n — An IEEE standard specifying data rates up to 250 Mbit/s and being backward compatible with 802.11a and 802.11g.

IEEE 802.16 — An IEEE standard specifying wireless last-mile broadband access. Also known as **WiMAX**.

IRLP — Internet Radio Linking Project, a system that uses the Internet to establish **VoIP** links among amateur stations who access the system via RF nodes at repeaters or on simplex frequencies.

***Linux* —** A free Unix-type operating system originated by Linus Torvalds, et al. Developed under the GNU General Public License.

MFSK — Multi-frequency shift keying.

***MFSK16* —** A multi-frequency shift communications system

***Modem* —** Modulator-demodulator, a device that connects between a data terminal and communication line (or radio).

***MSK* —** Frequency-shift keying where the shift in Hz is equal to half the signaling rate in bits per second.

***MT63* —** A keyboard-to-keyboard mode similar to PSK31 and RTTY.

Multiple protocol controller (MPC) — A piece of equipment that can act as a **TNC** for several **protocols**.

***Null modem* —** A device to interconnect two devices both wired as DCEs or DTEs; in EIA-232 interfacing, back-to-back DB25 connectors with pin-for-pin connections except that Received Data (pin 3) on one connector is wired to Transmitted Data (pin 3) on the other.

***Packet radio* —** A digital communications technique involving radio transmission of short bursts (frames) of data containing addressing, control and error-checking information in each transmission.

***PACTOR* —** Trade name of digital communications protocols offered by Special Communications Systems GmbH & Co KG (SCS).

***Parity check* —** Addition of non-information bits to data, making the number of ones in a group of bits always either even or odd.

Position encoder — A device that receives data from a GPS receiver, assembles APRS packets from the data and creates modulated signals for use by the transmitter.

***Project 25* —** Digital voice system developed for APCO, also known as P25.

***Protocol* —** A formal set of rules and procedures for the exchange of information within a network.

***PSK* —** Phase-shift keying.

***PSK31* —** A narrow-band digital communications system.

Radioteletype (RTTY) — Radio signals sent from one teleprinter machine to another machine. Anything that one operator types on his teleprinter will be printed on the other machine. Also known as narrow-band direct-printing telegraphy.

Sound card — A computer sound processing device that may be included on the motherboard, as a plug-in card, or as a separate external device.

Sound card interface — A device used to connect a computer sound card to a transceiver to provide TR switching and transmit/receive audio connections.

***Throb* —** A multi-frequency shift mode like MFSK16.

***TNC* —** Terminal node controller, a device that assembles and disassembles packets (frames); sometimes called a PAD.

Terminal unit (TU) — An interface used between an old teletype unit and a transceiver.

Turnaround time — The time required to reverse the direction of a half-duplex circuit, required by propagation, modem reversal and transmit-receive switching time of transceiver.

VoIP — Voice over Internet Protocol, a technology used to exchange audio information over the Internet.

Waterfall — A continuously scrolling display where the software processes and displays signatures of all signals detected within the bandwidth of the received audio signal.

WEP — Wired Equivalent Privacy. An encryption algorithm used by the authentication process for authenticating users and for encrypting data payloads over a WLAN.

WEP Key — An alphanumeric character string used to identify an authenticating station and used as part of the data encryption algorithm.

WiFi — Wireless Fidelity. Refers to products certified as compatible by the WiFi Alliance. See **www.wi-fi.org**. This term is also applied in a generic sense to mean any 802.11 capability.

***WiMAX* —** Familiar name for the IEEE 802.16 standard.

Wireless router — A low-power wireless device that manages data flow on a network that can include desktop or laptop computers, broadband Internet connections, printers and other compatible devices.

WIRES-II — Wide-Coverage Internet Repeater Enhancement System, a **VoIP** network created by Vertex Standard (Yaesu) that is functionally similar to **IRLP**.

WISP — Wireless Internet Service Provider

WLAN — Wireless Local Area Network.

WSJT — A suite of computer software for VHF/UHF digital communication including meteor scatter and moonbounce work.

12 Bibliography and References

BOOKS

R. Flickenger, *Building Wireless Community Networks* (O'Reilly & Associates, 2003).

S. Ford, WB8IMY, *ARRL VHF Digital Handbook* (ARRL, 2008).

S. Ford, WB8IMY, *ARRL HF Digital Handbook* (ARRL, 2007).

M. Gast; *802.11 Wireless Networks: The Definitive Guide* (O'Reilly & Associates, 2005).

B. Lafreniere, N6FN; *Nifty E-Z Guide to PSK31 Operation* (Nifty Ham Accessories, 2008).

D. Silage, *Digital Communication Systems Using* SystemVue (Charles River Media, 2006).

J. Taylor, K1RFD, *VoIP: Internet Linking for Radio Amateurs* (ARRL, 2009).

Radio Data Code Manual, (Klingenfuss; **www.klingenfuss.org**).

ARTICLES

S. Ford, WB8IMY, "Life Could Be a DReaM," *QST*, Apr 2007, pp 38-40.

M. Greenman, ZL1BPU, "MFSK for the New Millennium," *QST*, Jan 2001, pp 33-36.

G. Heron, N2APB "NUE-PSK Digital Modem," *QEX*, Mar/Apr 2008, pp 3-14.

S. Horzepa, WA1LOU, "Teaching an Old APRS New Tricks," *QST*, Feb 2006, pp 39-41.

M. Kionka, KIØGO, "Project 25 for Amateur Radio," *QST*, Sep 2008, pp 36-38.

P. Loveall, AE5PL "D-STAR Uncovered," *QEX*, Nov/Dec 2008, pp 50-56.

P. Martinez, G3PLX, "PSK31: A New Radio-Teletype Mode," *QEX*, Jul/Aug 1999, pp 3-9.

D. Schier, K1UHF, "The Ins and Outs of a Sound Card," *QST*, Oct 2003, pp 33-36.

J. Taylor, K1JT, "WSJT: New Software for VHF Meteor-Scatter Communication," *QST*, Dec 2001, pp 36-41.

J. Taylor K1RFD, "Computer Sound Cards for Amateur Radio," Product Review, *QST*, May 2007, pp 63-69.