V4 and V4Chat:

A Protocol and Client Optimized for Keyboard Radio QSOs

Rick Muething, KN6KB, AAA9WK (rmuething@cfl.rr.com)

Abstract:

The interest in digital modes and the growth of amateurs skilled in PC DSP programming has spurred development of application programs and protocols optimized for specific purposes. This paper describes examples of optimized client software and a sound card protocol designed specifically for keyboard and chat type QSO's including multi-user (NET) sessions. V4Chat and the V4 protocol team to provide hams with an easy to set up and operate keyboard mode with superior robustness that can handle 55 wpm typing speed while still fitting in a narrow 200 Hz bandwidth. V4 supports both ASCII and extended (foreign) UTF-8 character encoding and can operate in both FEC and ARQ modes.

Key Words:

V4, V4Chat, NVIS, Viterbi Encoded 4FSK, UTF-8, WINMOR

Motivation: Do we need yet another keyboard protocol?

During the development of WINMOR [1, 2] there were often requests for a robust, easy-to-use keyboard sound card mode that would "fit" into the narrow band segments of our HF bands. Although PSK 31 and other modes are very popular, poor propagation conditions (e.g. poor multipath as in NVIS propagation) sometime require more robust modulation and coding schemes to reach acceptable "copy". Robust FSK modes like MFSK16, DominoEX, THOR, and Olivia etc work well for NVIS but their bandwidth restricts them from some HF band segments. While WINMOR is primarily an ARQ (Automatic Retry reQuest) mode optimized for messages it does incorporate some unconnected (unproto) FEC capability for both 500 Hz and 1600 Hz bandwidths. However these WINMOR's unproto modes are not optimized for keyboard typing speeds (typically 40-60 words/minute) and the frame times (optimized for message forwarding) can lead to cumbersome keyboard turnaround time (reduced "slickness"). However the same robust modulation scheme used by WINMOR's unproto modes (Viterbi encoded 4FSK modulation) and the implementation of the virtual WINMOR TNC made it relatively easy to adapt a mode optimized for robustness, narrow bandwidth and the improved "slickness" desirable in keyboard applications. From this, the experimental new protocol V4 and a new keyboard client program V4Chat were developed to allow on-air testing this new protocol. The virtual TNC concept pioneered in WINMOR is also used to implement the virtual V4 TNC which allows other application developers easy access to and integration of the V4 protocol.

The goals of the V4 protocol are:

- Robust operation even in weak signal conditions and NVIS (multipath) propagation.
- Good bits/sec/Hz bandwidth efficiency capable in operating in narrow bandwidth restricted band segments.
- Support for modest typing speeds (40-60 words/minute).
- Support for domestic (ASCII) and extended (foreign) character sets. (UTF-8)
- Good "slickness" allowing for convenient type-to-send VOX-like exchanges.
- Rapid and automatic capture, tuning and tracking (+/- 100Hz).
- Suppress "empty channel" printing/display.
- The ability to work in both peer-to-peer or net configurations.
- Seamless support for both ARQ and non ARQ (FEC only) modes.
- Automatic CRC flagging to mark FEC text blocks containing uncorrectable errors.

Examples of the V4 protocol implementation:

The complete details of the V4 protocol are contained in the V4 specification [3]. A few of the highlights of the implementation are described here. While PSK modulation yields better bits/sec/Hz bandwidth efficiency it is quickly compromised in poor multipath propagation conditions like NVIS. FSK modes are more robust in these conditions but also take up more spectrum than PSK for similar throughput. A good compromise however is 4FSK where one of 4 tones (spaced at the symbol rate) is transmitted sending 2 bits/symbol. The bandwidth of this 4FSK is about 35 % more than QPSK for the same raw data rate at the – 26 dB points. To fit into a 200 Hz bandwidth 4FSK requires a symbol rate less than about 47 Hz. This translates to a raw uncoded bit rate of about 94 bits/second or about 117 words per minute based on 8 bit characters. But to achieve the degree of desired robustness some form of strong FEC (Forward Error Correction) must be used which of course reduces the net words per minute throughput. In PSK modulation a familiar and effective FEC mode is using QPSK to send 2 coded bits per symbol and to use a convolutional code and the efficient Viterbi decoding algorithm. A similar approach can be done for 4FSK using its two bits per symbol to send one user (information) bit per symbol using standard convolutional encoding (e.g. NASA Voyager R=1/2, K=7).

However the best performing Viterbi decoders use what are called "soft" inputs. These soft "1" and "0" inputs allow the Viterbi decoder to calculate a more accurate path through the decoding trellis which translates to about 2 dB of additional coding gain…the equivalent of increasing the transmitter power by 58%. This is enough motivation to look for a way to do this with the typical DSP based tone detectors commonly used to detect orthogonal FSK transmissions. Orthogonal detectors reject contributions by tones spaced at the baud rate. In V4 a novel mechanism based on a vector representation of the received tones is used to derive the soft inputs for the Viterbi decoder as summarized in Figure 1. Common DSP based tone detectors (FFT or Goertzel Algorithm) generate magnitude values (bins) for each of the 4 FSK frequencies F_0 through F_3. The 4 magnitudes are plotted along the 4 directions pointing to the "perfect" I and Q value pairs of the four corners. By

creating a normalized vector sum of the 4 magnitude vectors as indicated and projecting this sum onto the IQ plane we can translate the four received 4FSK frequency amplitude values to a pair of soft I and Q equivalent values for the Viterbi decoder. Using this approach we have an efficient 4FSK demodulator with soft outputs which can be fed directly to the Viterbi decoder. In this example 8 bit quantization of I and Q soft values are used (28=perfect 0, 228=perfect 1) which is more than sufficient.

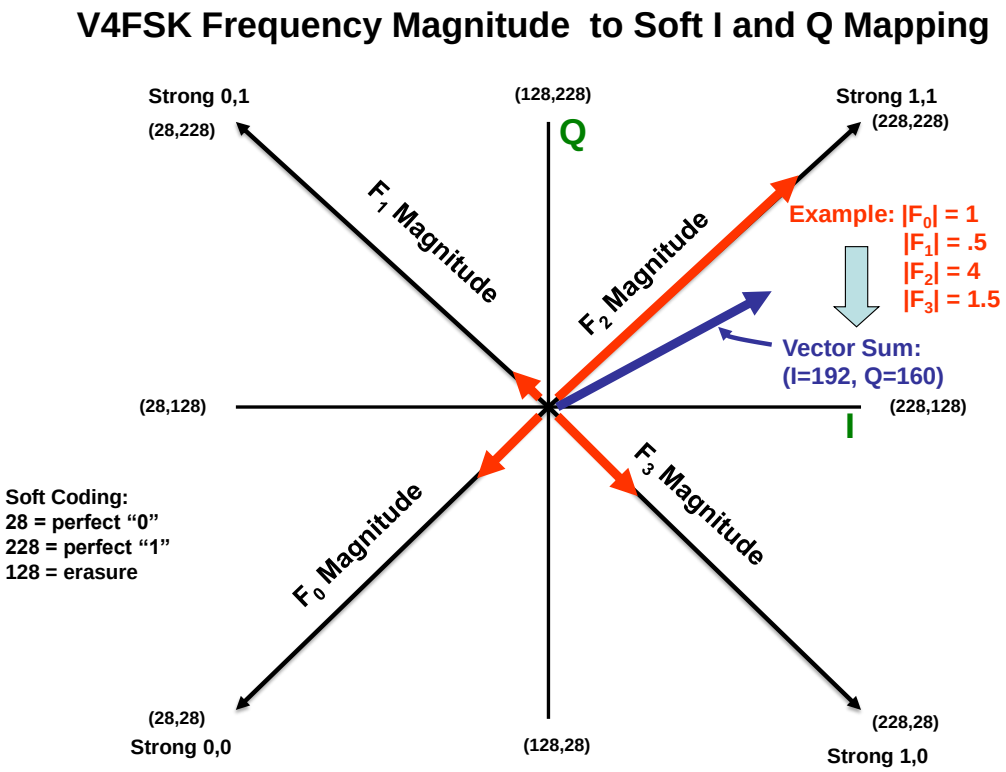

Figure 1 V4 Mapping of 4FSK magnitudes to soft IQ values for Viterbi Decoding

This additional coding gain however can't be fully realized unless the tone detectors are close to true orthogonal detectors. This either requires synchronous detection using some form of phase locking or precise alignment (tuning) to place the tones in the center of the DSP decoding bins (with adjacent tones in the bin's nulls). In V4 the technique used is to capitalize on a leader preceding each transmission block which communicates critical tuning information to the DSP decoder.

In V4 a short (170ms) leader is used at the start of each transmission block (16 characters). This leader has three important functions.

- Audio to trigger transmitter keying in VOX or VOX keyed Sound Card interfaces
- Tuning information allowing the receiving station(s) to lock on the V4 frame quickly (< 100 ms) and determine the frequency to within about 1 Hz accuracy.
- Provides initial framing information to establish symbol sync

To achieve this, an eight symbol (~170 ms) leader is generated using addition of the two center 4FSK tones to make a two-tone modulated signal as shown in Figure 2.

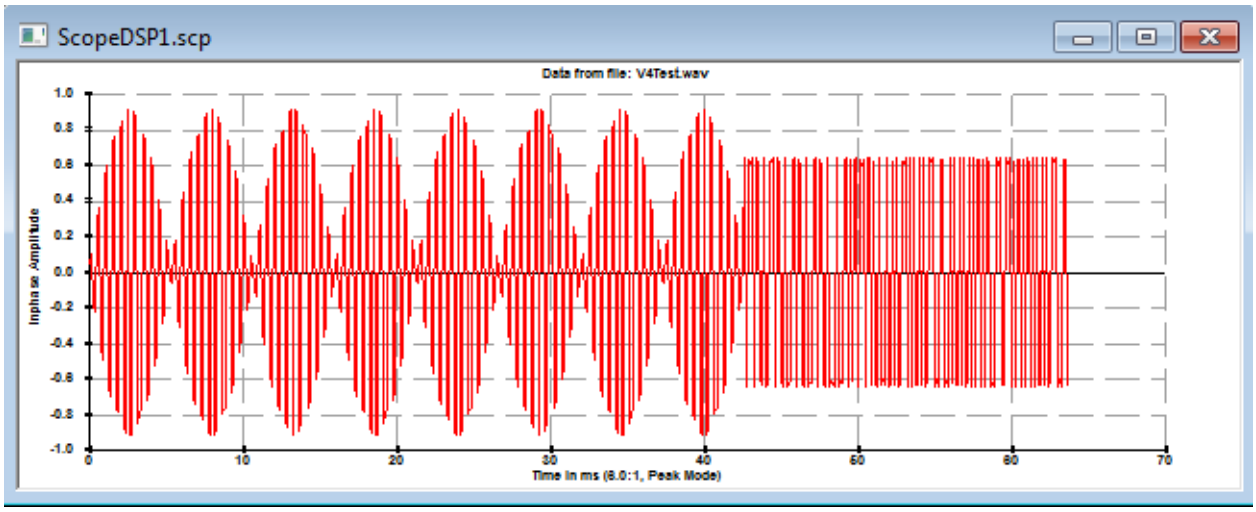

Figure 2 Waveform capture of the V4 Leader showing two-tone modulation preceding 4FSK.

Using a two-tone leader has specific advantages:

- It provides a unique frequency spectrum (different from 4FSK or a single tone) that permits detection and accurate interpolation of the frequencies to within about 1 Hz while rejecting single tone signals.
- It provides a waveform which can be envelope detected and filtered to obtain symbol sync. Figures 3 and 4 show how the leader envelope is used to establish initial symbol sync.

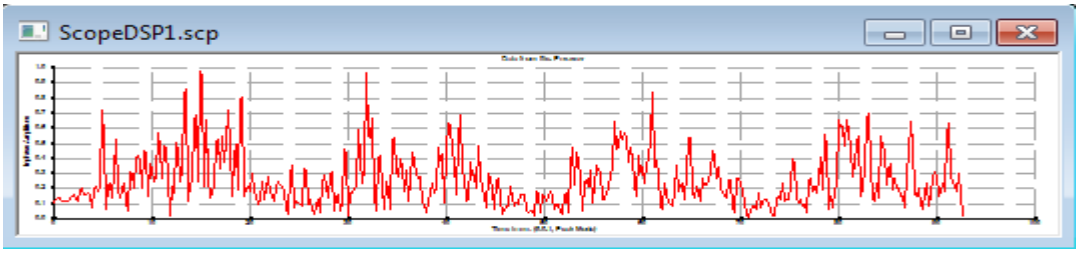

Figure 3. Leader envelope detection at -5dB S/N (3 KHz noise BW)

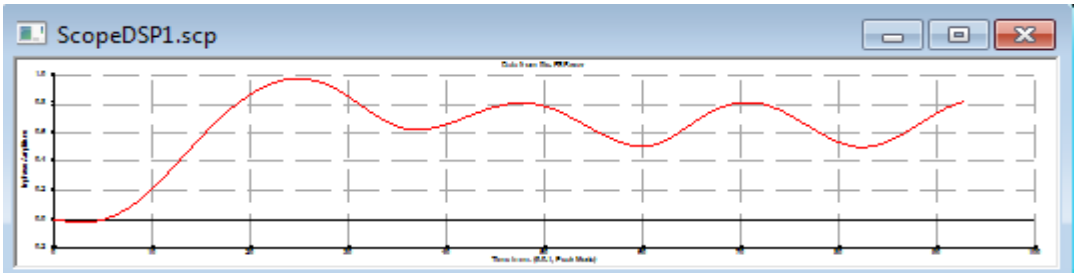

Figure 4. The filtered envelope of Fig 3 used to establish initial symbol sync

The output of the leader detector DSP code then tells the DSP software the precise center frequencies to use for each of the 4FSK Goertzel tone detectors and the initial symbol sync to begin 4FSK symbol decoding.

It is techniques such as these used in the V4 virtual TNC modem that give it both the robustness needed and the quick tuning that allow "slick" type-to-send keyboard operation.

Packaging the V4 protocol for distribution and integration

Developing a protocol and coding it are fine but to be useful that protocol needs to be adopted by both users and application developers. Asking developers to integrate complex real time DSP code into their applications is a burden. There are code and development system issues in addition to the time required to integrate and certify actual protocol implementation. For V4 I used a concept similar to what was used in WINMOR …the Virtual TNC. This is a stand-alone program that implements the protocol but communicates with the application through standard ports and commands…much like a conventional hardware packet or Pactor TNC does. The "front panel" for the V4 virtual TNC is shown in Figure 5. This is an actual capture of a full duplex FEC test at -5 dB S/N multipath poor channel (3 KHz Noise Bandwidth) using a HF channel simulator. The waterfall showing the 200 Hz bandwidth and the constellation diagram and score for the last frame are shown. You can think of the constellation diagram as a plot of all the received symbols of the frame …each pixel being the tip of the Sum Vector shown in Figure 1. The score is computed using the average distance of each symbol from its perfect I, Q value with a score of 100 being perfect (every symbol in a corner of the I, Q plane). In this marginal channel the symbols are poorly clustered yet still yield FEC decoding success over 62%. Using V4's ARQ mode in this channel would yield error-free communication at about 30 words per minute.

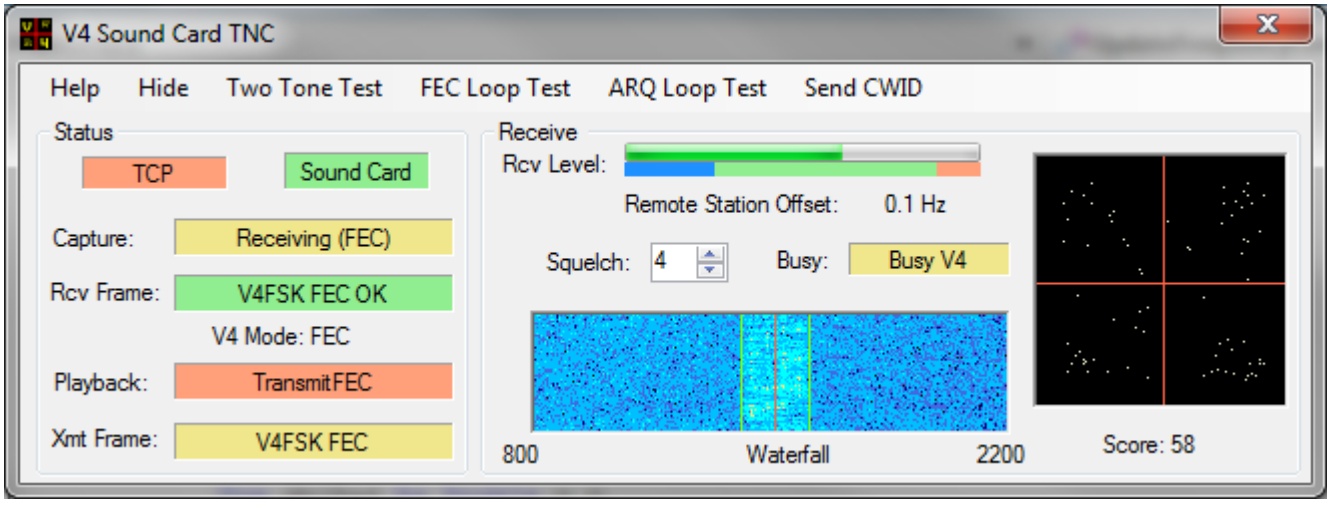

Figure 5. The V4 Virtual TNC "front panel"

To be useful however the virtual V4 TNC needs to be paired with an application program that provides a user interface, display, setup etc. This program also serves as a convenient test bed for the initial V4 on-air beta testing. An example of such a keyboard client program is V4Chat.

V4Chat: The Keyboard Client program for the V4 protocol

The simplified client program shown in Figure 6 provides the operator with convenient features for sending and receiving keyboard text and managing contacts. The program allows setting up the V4 virtual TNC parameters (call sign, sound card selection, drive level, etc.) and provides basic utilities for sending keyboard text or "pre canned" text files. The operator can select either FEC or ARQ transmit modes and allow either ARQ or simple FEC monitoring of received data. The use of rich text boxes, colored text and font styles are used to distinguish between sent text, Received ARQ or FEC text and to highlight FEC uncorrectable errors. All contacts are saved and date indexed for convenient retrieval or reference. The type-ahead buffer and quick turn over of the V4 protocol allows convenient type-to-send VOX-like operation either as a peer to peer QSO or in a NET. A useful feature for NETS is auto ID which precedes transmission blocks with the sender's call sign for easy identification.

Figure 6 V4Chat Keyboard Client and V4 Beta test vehicle

Summary

V4 should be an interesting and robust keyboard mode to experiment with and the V4 Virtual TNC should allow the protocol to be readily integrated into both new and existing client programs. Support for both ASCII and UTF-8 character encoding offers international appeal. The V4Chat client will evolve to a full function client with radio control and user requested features.

References

[1] WINMOR … A Sound Card ARQ Mode for Winlink HF Digital Messaging; Rick Muething, KN6KB/AAA9WK; 27th ARRL and TAPR DCC 2008

[2] WINMOR Phase 2: Demonstration to Deployment; Rick Muething, KN6KB/AAA9WK; 29th ARRL and TAPR DCC 2010

[3] For detailed specifications and links to software downloads go to www.winlink.org