Legacy Digital Modes

G-TOR

This brief description has been adapted from "A Hybrid ARQ Protocol for Narrow Bandwidth HF Data Communication" by Glenn Prescott, WBØSKX, Phil Anderson, WØXI, Mike Huslig, KBØNYK, and Karl Medcalf, WK5M (May 1994 *QEX*).

G-TOR is short for Golay-TOR, an innovation of Kantronics. It was inspired by HF automatic link establishment (ALE) concepts and is structured to be compatible with ALE. The purpose of the G-TOR protocol is to provide an improved digital radio communication capability for the HF bands. The key features of G-TOR are:

- Standard FSK tone pairs (mark and space)
- Link-quality-based signaling rate: 300, 200 or 100 baud
- 2.4-s transmission cycle
- Low overhead within data frames
- Huffman data compression — two types, on demand
- Embedded run-length data compression
- Golay forward-error-correction coding
- Full-frame data interleaving
- CRC error detection with hybrid ARQ
- Error-tolerant "Fuzzy" acknowledgments.

Since one of the objectives of this protocol is ease of implementation in existing TNCs, the modulation format consists of standard tone pairs (FSK), operating at 300, 200 or 100 baud, depending upon channel conditions. G-TOR initiates contacts and sends ACKs only at 100 baud. The G-TOR waveform consists of two phase-continuous tones (BFSK), spaced 200 Hz apart (mark = 1600 Hz, space = 1800 Hz); however, the system can still operate at the familiar 170 Hz shift (mark = 2125 Hz, space = 2295 Hz), or with any other convenient tone pairs. The optimum spacing for 300-baud transmission is 300 Hz, but you trade some performance for a narrower bandwidth.

Each transmission consists of a synchronous ARQ 1.92-s frame and a 0.48-s interval for propagation and ACK transmissions (2.4 s cycles). All advanced protocol features are implemented in the signal-processing software.

Data compression is used to remove redundancy from source data. Therefore, fewer bits are needed to convey any given message. This increases data throughput and decreases transmission time — valuable features for HF. G-TOR uses run-length encoding and two types of Huffman coding during normal text transmissions. Run-length encoding is used when more than two repetitions of an 8-bit character are sent. It provides an especially large savings in total transmission time when repeated characters are being transferred.

The Huffman code works best when the statistics of the data are known. G-TOR applies Huffman A coding with the upper- and lower-case character set, and Huffman B coding with upper-case-only text. Either type of Huffman code reduces the average number of bits sent per character. In some situations, however, there is no benefit from Huffman coding. The encoding process is then disabled. This decision is made on a frame-by-frame basis by the information sending station.

The real power of G-TOR resides in the properties of the (24, 12) extended Golay error-correcting code, which permits correction of up to three random errors in three received bytes. The (24, 12) extended Golay code is a half-rate error-correcting code: Each 12 data bits are translated into an additional 12 parity bits (24 bits total). Further, the code can be implemented to produce separate input-data and parity-bit frames.

The extended Golay code is used for G-TOR because the encoder and decoder are simple to implement in software. Also, Golay code has mathematical properties that make it an ideal choice for short-cycle synchronous communication. More information can also be found online at **www.arrl.org/technical-characteristics**.

CLOVER-II

The desire to send data via HF radio at high data rates and the problem encountered when using AX.25 packet radio on HF radio led Ray Petit, W7GHM, to develop a unique modulation waveform and data transfer protocol that is now called CLOVER-II. Bill Henry, K9GWT, supplied this description of the CLOVER-II system.

CLOVER modulation is characterized by the following key parameters:

- Very low base symbol rate: 31.25 symbols/second (all modes).
- Time-sequence of amplitude-shaped pulses in a very narrow frequency spectrum.
- Occupied bandwidth = 500 Hz at 50 dB below peak output level.
- Differential modulation between pulses.
- Multilevel modulation.

The low base symbol rate is very resistant to multipath distortion because the time between modulation transitions is much longer than even the worst-case time-smearing caused by summing of multipath signals. By using a time-sequence of tone pulses, Dolph-Chebychev "windowing" of the modulating signal and differential modulation, the total occupied bandwidth of a CLOVER-II signal is held to 500 Hz.

Multilevel tone, phase and amplitude modulation gives CLOVER a large selection of data modes that may be used (see **Table 1**). The adaptive ARQ mode of CLOVER senses current ionospheric conditions and automatically adjusts the modulation mode to produce maximum data throughput. When using the Fast bias setting, ARQ throughput automatically varies from 11.6 byte/s to 70 byte/s.

The CLOVER-II waveform uses four tone pulses that are spaced in frequency by 125 Hz. The time and frequency domain characteristics of CLOVER modulation are shown in **Figure 1**, **2** and **3**. The time-domain shape of each tone pulse is intentionally shaped to produce a very compact frequency spectrum. The four tone pulses are spaced in time and then combined to produce the composite output

Table 1
CLOVER-II Modulation Modes

As presently implemented, CLOVER-II supports a total of seven different modulation formats: five using PSM and two using a combination of PSM and ASM (Amplitude Shift Modulation).

Name	*Description*	*In-Block Data Rate*
16P4A	16 PSM, 4-ASM	750 bps
16PSM	16 PSM	500 bps
8P2A	8 PSM, 2-ASM	500 bps
8PSM	8 PSM	375 bps
QPSM	4 PSM	250 bps
BPSM	Binary PSM	125 bps
2DPSM	2-Channel Diversity BPSM	62.5 bps

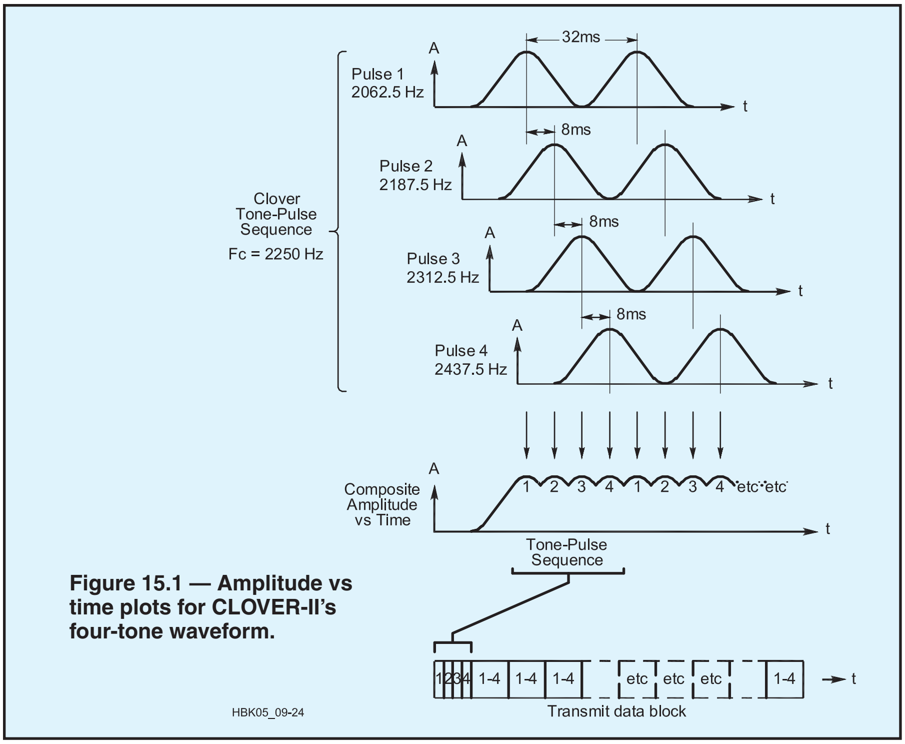

Figure 15.1 — Amplitude vs time plots for CLOVER-II's four-tone waveform.

Table 2

Data Bytes Transmitted Per Block

Block Size	*Reed-Solomon Encoder Efficiency 60%*	*75%*	*90%*	*100%*
17	8	10	12	14
51	28	36	42	48
85	48	60	74	82
255	150	188	226	252

Table 3

Correctable Byte Errors Per Block

Block Size	*Reed-Solomon Encoder Efficiency 60%*	*75%*	*90%*	*100%*
17	1	1	0	0
51	9	5	2	0
85	16	10	3	0
255	50	31	12	0

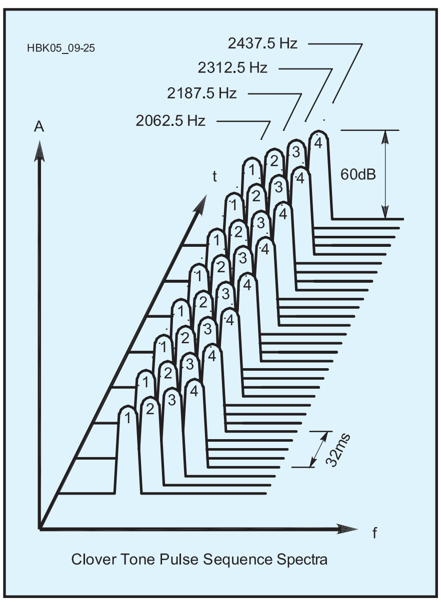

Figure 2 — A frequency-domain plot of a CLOVER-II waveform.

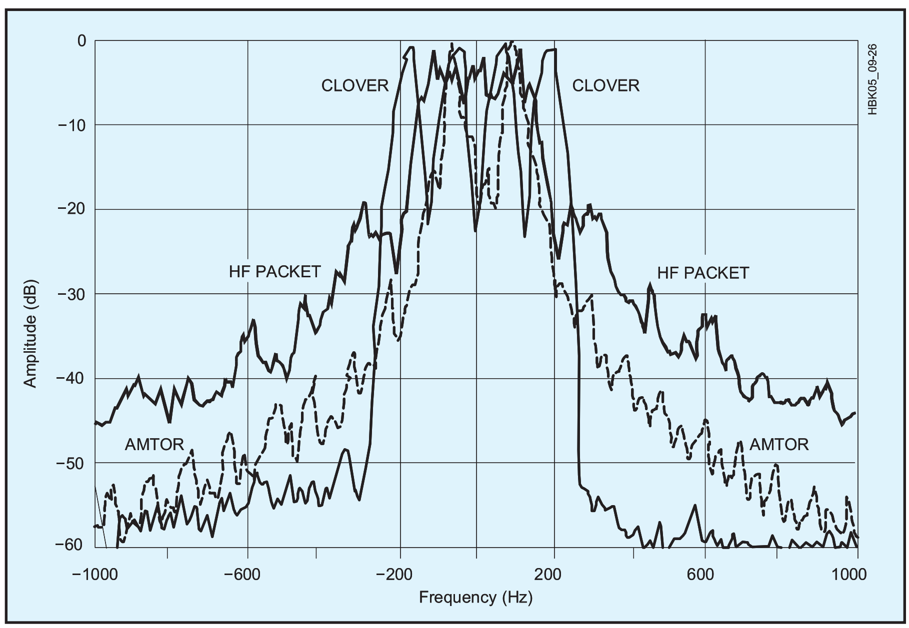

Figure 3 — Spectra plots of AMTOR, HF packet-radio and CLOVER-II signals.

shown. Unlike other modulation schemes, the CLOVER modulation spectrum is the same for all modulation modes.

Data is modulated on a CLOVER-II signal by varying the phase and/or amplitude of the tone pulses. Further, all data modulation is differential on the same tone pulse — data is represented by the phase (or amplitude) difference from one pulse to the next. For example, when binary phase modulation is used, a data change from 0 to 1 may be represented by a change in the phase of tone pulse one by 180° between the first and second occurrence of that pulse. Further, the phase state is changed only while the pulse amplitude is zero. Therefore, the wide frequency spectra normally associated with PSK of a continuous carrier is avoided. This is true for all CLOVER-II modulation formats. The term *phase-shift modulation* (PSM) is used when describing CLOVER modes to emphasize this distinction.

CLOVER-II has four "coder efficiency" options: 60%, 75%, 90% and 100% ("efficiency" being the approximate ratio of real data bytes to total bytes sent). 60% efficiency corrects the most errors but has the lowest net data throughput. 100% efficiency turns the encoder off and has the highest throughput but fixes no errors. There is therefore a tradeoff between raw data throughput versus the number of errors that can be corrected without resorting to retransmission of the entire data block.

Note that while the In Block Data Rate numbers listed in the table go as high as 750 bit/s, overhead reduces the net throughput or overall efficiency of a CLOVER transmission. The FEC coder efficiency setting and protocol requirements of FEC and ARQ modes add overhead and reduce the net efficiency. **Tables 2** and **3** detail the relationships between block size, coder efficiency, data bytes per block and correctable byte errors per block.

With seven different modulation formats, four data-block lengths (17, 51, 85 or 255 bytes) and four Reed-Solomon coder efficiencies (60%, 75%, 90% and 100%), there are 112 (7 × 4 × 4) different waveform modes that could be used to send data via CLOVER. Once all of the determining factors are considered, however, there are eight different waveform combinations that are actually used for FEC and/or ARQ modes.

CLOVER-2000

CLOVER-2000 is a faster version of CLOVER (about four times faster) that uses eight tone pulses, each of which is 250 Hz wide, spaced at 250 Hz centers, contained within the 2 kHz bandwidth between 500 and 2500 Hz. The eight tone pulses are sequential, with only one tone being present at any instant and each tone lasting 2 ms. Each frame consists of eight tone pulses lasting a total of 16 ms, so the base modulation rate of a CLOVER-2000 signal is always 62.5 symbols per second (regardless of the type of modulation being used). CLOVER-2000's maximum raw data rate is 3000 bit/s.

Allowing for overhead, CLOVER-2000 can deliver error-corrected data over a standard HF SSB radio channel at up to 1994 bit/s, or 249 characters (8-bit bytes) per second. These are the uncompressed data rates; the maximum throughput is typically doubled for plain text if compression is used. The effective data throughput rate of CLOVER-2000 can be even higher when binary file transfer mode is used with data compression.

The binary file transfer protocol used by HAL Communications operates with a terminal program explained in the HAL E2004 engineering document. Data compression algorithms tend to be context sensitive — compression that works well for one mode (say, text), may not work well for other data forms (graphics, for example). The HAL terminal program uses the PK-WARE compression algorithm, which has proved to be a good general-purpose compressor for most computer files and programs. Other algorithms may be more efficient for some data formats, particularly for compression of graphic image files and digitized voice data. The HAL Communications CLOVER-2000 modems can be operated with other data compression algorithms in the users' computers.

CLOVER-2000 is similar to the previous version of CLOVER, including the transmission protocols and Reed-Solomon error detection and correction algorithm. The original descriptions of the CLOVER Control Block (CCB) and Error Correction Block (ECB) still apply for CLOVER-2000, except for the higher data rates inherent to CLOVER-2000. Just like CLOVER, all data sent via CLOVER-2000 is encoded as 8-bit data bytes and the error-correction coding and modulation formatting processes are transparent to the data stream — every bit of source data is delivered to the receiving terminal without modification.

Control characters and special "escape sequences" are not required or used by CLOVER-2000. Compressed or encrypted data may therefore be sent without the need to insert (and filter) additional control characters and without concern for data integrity. Five different types of modulation may be used in the ARQ mode — BPSM (Binary Phase Shift Modulation), QPSM (Quadrature PSM), 8PSM (8-level PSM), 8P2A (8PSM + 2-level Amplitude-Shift Modulation) and 16P4A (16 PSM plus 4 ASM).

The same five types of modulation used in ARQ mode are also available in Broadcast (FEC) mode, with the addition of 2-Channel Diversity BPSM (2DPSM). Each CCB is sent using 2DPSM modulation, 17-byte block size and 60% bias. The maximum ARQ data throughput varies from 336 bit/s for BPSM to 1992 bit/s for 16P4A modulation. BPSM is most useful for weak and badly distorted data signals, while the highest format (16P4A) needs extremely good channels, with high SNRs and almost no multipath.

Most ARQ protocols designed for use with HF radio systems can send data in only one direction at a time. CLOVER-2000 does not need an OVER command; data may flow in either direction at any time. The CLOVER ARQ time frame automatically adjusts to match the data volume sent in either or both directions. When first linked, both sides of the ARQ link exchange information using six bytes of the CCB. When one station has a large volume of data buffered and ready to send, ARQ mode automatically shifts to an expanded time frame during which one or more 255 byte data blocks are sent.

If the second station also has a large volume of data buffered and ready to send, its half of the ARQ frame is also expanded. Either or both stations will shift back to CCB level when all buffered data has been sent. This feature provides the benefit of full-duplex data transfer but requires use of only simplex frequencies and half-duplex radio equipment. This two-way feature of CLOVER can also provide a back-channel "order-wire" capability. Communications may be maintained in this chat mode at 55 WPM, which is more than adequate for real-time keyboard-to-keyboard communications.

More information can also be found at **www.arrl.org/technical-characteristics**.

WINMOR

While the various PACTOR modes currently dominate and generally represent the best available performance HF ARQ protocols suitable for digital messaging, PC sound cards with appropriate DSP software can now begin to approach PACTOR performance. The WINMOR (Winlink Message Over Radio) protocol is an outgrowth of SCAMP (Sound Card Amateur Message Protocol) by Rick Muething, KN6KB. SCAMP put an ARQ "wrapper" around Barry Sanderson's RDFT (Redundant Digital File Transfer) then integrated SCAMP into a Client and Server for access to the Winlink message system. (More on Winlink in a later section.) SCAMP worked well on good channels but suffered from the following issues:

- The RDFT batch-oriented DLLs were slow and required frame pipelining, increasing complexity and overhead.
- RDFT only changed the RS encoding on its 8PSK multi carrier waveform to achieve a 3:1 range in speed/robustness which is not enough.
- RDFT was inefficient in Partial Frame recovery (no memory ARQ).
- RDFT was a 2.4 kHz mode and limited to narrow HF sub bands.
- SCAMP's simple multi-tone ACK/NAK did not carry session ID info, increasing chances of fatal cross session contamination.

WINMOR is an ARQ mode generated from the ground up to address the limitations of SCAMP/RDFT and leverage what was learned. Today, a viable message system (with the need for compression and binary attachments) requires true "error-free" delivery of binary data. To achieve this there must be some "back channel" or ARQ so the receiving station can notify the sender of lost or damaged data and request retransmission or repair. **Table 4** outlines the guidelines used in the development of WINMOR.

Perhaps the most challenging of these requirements are:

- The ability to quickly tune, lock and acquire the signal which is necessary for practical length ARQ cycles in the 2-6 s range.

Table 15.4
WINMOR Development Guidelines

Absolute Requirements	*Desirable Requirements*
Work with standard HF (SSB) radios	Modest CPU & OS demands
Accommodate Automatic Connections	Bandwidth options (200, 500, 3000 Hz)
Error-free transmission and confirmation	Work with most sound cards/interfaces
Fast Lock for practical length ARQ cycles	Good bit/s/Hz performance ~ P2 goal
Auto adapt to a wide range of changing channel conditions	Efficient modulation and demodulation for acceptable ARQ latency
Must support true transparent binary to allow attachments and compression	Selective ARQ & memory ARQ to maximize throughput and robustness.
Must use loosely synchronous ARQ timing to accommodate OS and DSP demands	Near Pactor ARQ efficiency (~70% of raw theoretical throughput)

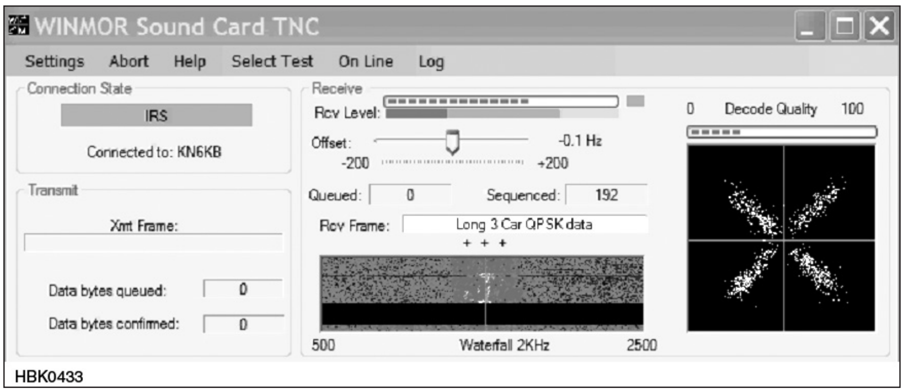

Figure 4 — WINMOR sound card TNC screen.

- The ability to automatically adapt the modulation scheme to changing channel conditions. An excellent example of this is Pactor III's extremely wide range of speed/robustness (18:1) and is one reason it is such an effective mode in both good and poor channel conditions.

The most recent development effort has focused on 62.5 baud BPSK, QPSK and 16QAM and 31.25-baud 4FSK using 1 (200 Hz), 3 (500 Hz) and 15 (2000 Hz). With carriers spaced at twice the symbol rate. These appear to offer high throughput and robustness especially when combined with multi-level FEC coding.

WINMOR uses several mechanisms for error recovery and redundancy.

1) FEC data encoding currently using:

- 4,8 Extended Hamming Dmin = 4 (used in ACK and Frame ID)
- 16-bit CRC for data verification

Two-level Reed-Solomon (RS) FEC for data:

- First level Weak FEC, for example RS 140,116 (corrects 12 errors)
- Second level Strong FEC, for example RS 254,116 (corrects 69 errors)

2) Selective ARQ. Each carrier's data contains a Packet Sequence Number (PSN). The ACK independently acknowledges each PSN so only carriers with failed PSNs get repeated. The software manages all the PSN accounting and re-sequencing.

3) Memory ARQ. The analog phase and amplitude of each demodulated symbol is saved for summation (phasor averaging) over multiple frames. Summation is cleared and restarted if max count reached. Reed-Solomon FEC error decoding done after summation.

4) Multiple Carrier Assignment (MCA). The same PSN can be assigned to multiple carriers (allows tradeoff of throughput for robustness). Provides an automatic mechanism for frequency redundancy and protection from interference on some carriers.

5) Dynamic threshold adjustment (used on QAM modes) helps compensate for fading which would render QAM modes poor in fading channels.

In trying to anticipate how WINMOR might be integrated into applications they came up with a "Virtual TNC" concept. This essentially allows an application to integrate the WINMOR protocol by simply treating the WINMOR code as just another TNC and writing a driver for that TNC. Like all TNCs there are some (<10) parameters to set up: call sign, timing info, sound card, keying mechanism, etc. A sample image of the virtual TNC appears in **Figure 4**.

The WINMOR software DLL can even be made to appear as a physical TNC by "wrapping" the DLL with code that accesses it through a virtual serial port or a TCP/IP port. Like a physical TNC WINMOR has a "front panel" with flashing lights. But since operation is automatic with no front panel user interaction required the WINMOR TNC can be visible or hidden.

WINMOR looks promising and the testing to date confirms:

- Sound card ARQ is possible with a modern CPU and OS while making acceptable CPU processing demands. (CPU Loading of < 20% on a 1.5 GHz Celeron/Win XP)
- Throughput and robustness can be adjusted automatically to cover a wide range of bandwidth needs and channel conditions. (10:1 bandwidth range, 57:1 throughput range)
- ARQ throughput in excess of 0.5 bit/s/Hz is possible in fair to good channels (0.68 – 0.82 bit/s/Hz measured)
- Good ARQ efficiency — 70-75%
- Throughput is currently competitive with P2 and P3 and significantly better than P1

More information about WINMOR is available at **www.winlink.org/Winlink Express**.