Throughput and Probability of Correct Message Transfer of the Pactor-II System Measured on Near-Vertical-Incidence-Skywave (NVIS) Paths

Ken Wickwire (KB1JY), Mike Bernock (KB1PZ) and Dave Willard (W1EO)

1. Introduction

This paper is another in our series treating on-air measurement of throughput in characters per second (cps) for various HF data-transmission protocols of interest to amateurs (see the references to our other reports at the end of the paper). Here we describe an extensive set of measurements of throughput for compressed and uncompressed text files sent over near-vertical-incidence-skywave (NVIS) paths with the Pactor-II data transfer protocol. The implementation we used was the one in the Special Communications Systems (SCS) modem.

NVIS paths, which often experience relatively difficult channel conditions characterized by multipath interference, high noise, nighttime QRM and midday absorption by the so-called D-Layer, are used to communicate over roughly 20- to 300-mile ground distances using antennas that can launch energy at high takeoff angles (horizontal dipoles, sloping longwires, bent-over whips, etc.). Since conditions on NVIS links are almost always worse than those on longer one-hop skywave links, evaluations of performance on NVIS links provide conservative (lower bound) predictions of skywave performance.

Despite its relatively high price ($650-$950), the Pactor-II system has become popular in amateur circles as an important hardware component of the admirable Winlink and Airmail HF E-mail systems. These systems are used by hams in sailboats and RVs to keep in touch with fixed (often non-amateur) sites via a combination of HF and the Internet. Two versions of the Pactor-II modem are available, the PTC-II and the PTC-IIe. These differ mainly in the “non-Pactor-II” features (packet modem slots, rig control interface, text display on modem front panel) that come with the PTC-II but not with the PTC-IIe. All of our testing used PTC-IIs with firmware versions 2.7 or later installed (the newest firmware as of this writing is version 3.0).

Pactor-II is an improvement of the widely used Pactor-I frequency shift keying (FSK) system. Pactor-II uses a two-tone, differential phase-shift keyed (DPSK) waveform whose signal-constellation size is chosen from among two, four, eight and sixteen phase shifts (DBPSK, DQPSK, 8-DPSK and 16-DPSK). The corresponding raw (channel) data rates are 200, 400, 600 and 800 bits per second (bps).

Error control is accomplished in two ways in Pactor-II: with forward error correction (FEC) and with an automatic repeat request (ARQ) scheme. (This powerful combination is employed by almost every modern point-to-point HF data communications system.) The FEC uses an interleaver, a constraint-length-9 convolutional encoder and a Viterbi soft-decision decoder.

As the Pactor-II protocol adapts itself to changing channel conditions in ARQ (connected) operation, it chooses its modulation mode from those described above, along with its FEC coding rate, which is the ratio of data to FEC coding bits in data packets. These two kinds of change are coupled, so that as the size of the signaling constellation goes up (when the protocol determines that conditions are good), the coding rate also goes up (i.e., less coding overhead is sent with each packet than is sent in poorer conditions). This has the effect of pushing more data through the channel when it’s good

and slowing things down when there are many bit errors and many frame repeats are requested by a receiving station. The net result of the changes is achievement of roughly the highest protocol throughput the channel will support at any particular time.

As with most other commercial adaptive HF data communications systems, details on precisely what the Pactor-II protocol measures to adapt itself to channels are proprietary. The same is true of the algorithms that do the adaptation, since such algorithms take a great deal of thought and experimentation to perfect.

The Pactor-II ARQ scheme takes advantage of the soft (analog) decisions made by the FEC decoder to perform "memory ARQ." In this technique, a data packet corrupted by so many channel errors that the FEC scheme can't correct it is saved and subsequently combined with other repeats of the same packet to reconstruct an error-free version of the original sent packet. All packets received in the ARQ mode are checked for errors with a relatively powerful 16-bit cyclic redundancy check (CRC) sequence.

As a further throughput-enhancing device the system uses various (settable) modes of data-compression, applied on a packet-by-packet basis. There are three compression techniques. These are run-length, Huffman and "pseudo-Markov coding" (PMC). Run-length compression is used when there are long series of repeated bytes in a file; for example, underscores. Huffman compression uses language-specific character-occurrence statistics to assign short bit-codes to characters (like "e" in English texts) that appear frequently. PMC is an advanced compression technique that codes pairs of commonly occurring characters rather than single characters, as in the Huffman approach.

The compression modes are

- Mode 0: 8-bit ASCII (no compression).
- Mode 1: 7-bit ASCII with Huffman compression or 8-bit ASCII, whichever makes the message smaller.
- Mode 2: Run length, Huffman and PMC, or 8-bit ASCII when required.

Compression is generally a good idea when sending files in the "connected" (ARQ) mode. Exceptions are executable and certain already-compressed (e.g., graphics) files, which sometimes actually get bigger when compression algorithms are applied to them. Compression is not normally used in the broadcast (non-ARQ) mode since in that mode receivers have no way of asking for repeats of frames that arrive with corrupted compression information. To get a good understanding of the average throughput in each compression mode we gathered a large amount of data in each mode.

Various user interfaces are available for operating Pactor-II modems, including those that allow one to send graphics and other "binary" (eight-bit-character) file data, in addition to text files. To send files in our testing we wrote C-code that interfaced PCs and Macs directly with the modems using the modems' built-in command set. We received files in the Pactor-II and Airmail mailboxes.

The Airmail mailbox resides on PCs running the Airmail software. The Pactor-II mailbox is in the Pactor-II modem itself. We treated the two mailbox systems separately because we did not know at the outset of our testing if the two systems would produce the same performance in the same conditions. (More on this later.) For further details on how Pactor-II works, see the documentation supplied with the modems and at the voluminous SCS website (URL: http://www.scs-ptc.com).

The NVIS stations we used for our measurements are 30 to 110 miles apart, and are located in Massachusetts, New Hampshire and Maine. NVIS paths often display strong multipath, high local and propagated noise, D-layer absorption at mid-day and occasionally strong interference from other stations operating in both voice and digital modes. (Horizontal antenna polarization at all our NVIS stations allowed us to be pretty confident that we were using NVIS rather than surfacewave propagation, and this was confirmed by the fading and other skywave propagation phenomena we observed during numerous file transfers.)

We tested Pactor-II in the ham bands and also on assigned frequencies outside the ham bands. The frequencies used were between 3 and 10 MHz, with frequencies in the lower half of this range being used a night. Since the average sunspot number was high (over 100) during most of the test period, solar flares and other magnetic activity were relatively common. We were thus able to enjoy the higher frequencies that come with increased sunspot activity while also observing near or complete propagation blackouts caused by solar flares. It was an interesting time for NVIS testing. The tests covered the period from June 1999 to August 2000.

We used standard Yaesu (FT-1000MP, FT-757GX) and Icom (IC-751A) amateur transceivers, all running about 100 watts. On radios equipped for it (the 1000MPs) we used 500-Hz receiving filters. A discussion of the antennas we used is in the next section. We ran tests during both day- and nighttime using automated testing and data-logging software we wrote for that purpose.

Daytime was defined to occur between the fixed times of 1000 and 2200 GMT (5:00 AM to 5:00 PM local time). Note that because our measurements were made over the course of more than a year, “nighttime” (5:00 PM to 5:00 AM local time) was not always associated with darkness at the path midpoint. Nevertheless, most nighttime measurements were taken in conditions that characterize conventional nighttime HF propagation: high noise and increased interference.

Frequencies ten or twenty percent below the so-called Maximum Usable Frequency (MUF) are traditionally thought to be the best ones to use for HF. Since we achieved the goal of working just below the MUF only approximately with our available frequencies, our results are conservative. A properly set up automatic link establishment (ALE) system that used quasi-real-time channel assessments to choose the best operating channel for the Pactor-II waveform might have allowed us to avoid the “far-from-the-MUF” conditions we occasionally encountered and thus achieve higher throughput. We say “might” because Mil-Std-188-141A ALE and Pactor-II use different waveforms. The extent to which channel-quality measurements made with FSK ALE can predict the best frequency to use with PSK Pactor-II is not clear.

The rest of the paper describes the paths between stations and antennas, the automated test software and our file-transfer operations, the recorded data format, our special-purpose statistical analysis software, a statistical summary of the data, a discussion of the statistical results and concluding remarks.

2. Layout of Paths and Discussion of Antennas

The stations used for the NVIS tests are in Bedford, Mass. (KB1JY/BED1), Derry, N.H. (KB1PZ/DER1), and Portland, Maine (KB1JY-1/POR1). Figure 1 gives the layout of the stations. Bedford used 80m dipoles up 30 feet. Derry used an off-center-fed 80m dipole up 30 feet or a terminated sloping longwire about 180 feet long. Portland used an end-fed

120-foot unterminated longwire pointed southwest. The links (followed by lengths and rough estimates of the percentage of data collected over each link) are

NVIS Links

- Bedford-Derry (35 miles, 30%)
- Portland-Derry (65 miles, 30%)
- Portland-Bedford (95 miles, 40%)

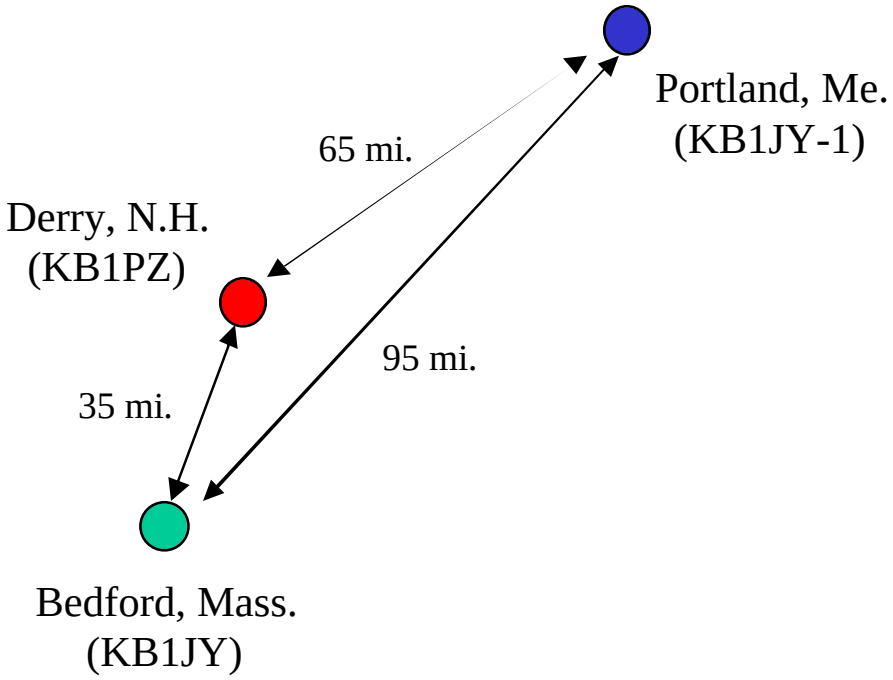

Fig. 1. Layout of Our NVIS Stations

3. The Automated Test Programs and Test Flow

To automate our testing we wrote two programs that connect to the Pactor-II and Airmail mailboxes, send the appropriate send-message command and message subject at the mailbox prompt, upload a canned text message of known size, close the link and log various statistics of the transfer to an archive and a log file. (Two programs were necessary because of minor differences in the prompts issued by each of the mailboxes.)

Since we used both PCs and Macs in our testing we actually wrote a total of four test programs. (Their modular construction makes them look very similar.) The PCs ran 16-bit executables in DOS console windows and the Macs ran standard console applications. The corresponding pairs of PC and Mac programs look almost identical. Only the routines for setting up comports and for character I/O were different since the two operating systems use different approaches to serial-port I/O (direct access to hardware and do-it-yourself interrupt handling in DOS and the Toolbox serialcom API for everything on the Mac).

The archive contains one line of performance data for each transfer. The data in these lines are discussed below. The archive also contains data-lines for transfers that timed out (failed); such lines are used to calculate transfer-success probabilities. The log file (written also to the console for monitoring of test progress) contains details of each message transmission and is normally overwritten by log data for the next transfer. We used data in the log file to fine-tune the testing software and occasionally to shed light on anomalous protocol behavior.

Shown below is a log file for a successful transfer of a 25k text file to a Pactor-II mailbox. The file contains a time-tagged narrative of what happened during the transfer and concludes with performance statistics that are similar to those that go into the performance archive file. Characters in square brackets in the log file are those sent by the local (file-sending) Pactor-II modem to the local PC running the test program. This is the character stream that is monitored by the test program for events that trip actions like sending a message-subject string or recording the message transfer time. Square-bracketed characters appearing after the first set have been suppressed to save space.

No xfers = 1 Time between xfers = 1 s
PTC-II TRANSFER(S) @ BED1 TO PTC-II MBOX v. macPTC26 Mode = 2
30.06.00 15:52:09 C DER1

[0xD][0xA][c][m][d][:][ ][0xD][0xA][*][*][*][ ][C][L][R][0xD][0xA][0xD][0xA][c][m]
[d][:][ ][0xD][0xA][*][*][*][-][P][A][C][T][O][R][][T][R][A][N][S][M][I][S][S][I][O][N]
[-][M][O][D][E][ ][(][0][=][A][S][C][I][I][/][1][=][H][U][F][F][M][A][N][/][2][=][P][M]
[C][)][:][ ][2][0xD][0xA][0xD][0xA][c][m][d][:][ ][0x1E][0xD][0xA][*][*][*][ ][N][O]
[W][ ][C][A][L][L][I][N][G][ ][D][E][R][1][0xD][0xA][0x1E][0x1E]
[*][*][*][C][O][N][N][E][C][T][E][D]
30.06.00 15:52:11 Connected

30.06.00 15:52:20 BBS prompt rcvd
30.06.00 15:52:20 S BED1 file.25k

30.06.00 15:52:33 Subject prompt rcvd, sending file.25k...

msg file size = 25000 uploads = 3 last_upload = 6006
30.06.00 15:52:45 0xFFA4 slave CHO
30.06.00 15:52:45 0xFFAA master traffic
30.06.00 15:52:45 0xFFA9 master request
30.06.00 15:52:45 0xFFAA master traffic
…
30.06.00 15:55:30 rem read 1 10000 chars sent, 9000 rem-echoed chars rcvd
…
30.06.00 15:55:58 0xFFA9 master request
30.06.00 15:56:02 0xFFAA master traffic
30.06.00 15:56:06 0xFFA8 master error
…
30.06.00 15:58:36 rem read 2 19000 chars sent, 18000 rem-echoed chars rcvd
30.06.00 15:58:44 Sending EOM
…
30.06.00 16:00:07 rem read 3 25000 chars sent, 25000 rem-echoed chars retrieved
…
30.06.00 16:00:10 Message "file.25k" filed.
LinkTime 2 s NegotiationTime 22 s

25006 bytes sent in 456 s; 54.8 bytes/s 35 TxARQs 1 TxERRs
30.06.00 16:00:10 BYE sent...
…
30.06.00 16:00:21 Disconnected
Trial 1: Successful xfer to PTC-II MBOX

The log starts by recording the number of scheduled message transfers (here one) and the time between transfers (entered here as one second but ignored since only one transfer is scheduled). This is followed by a banner announcing which mailbox system is being called, the software version and the compression mode (Mode = 0, 1 or 2; see Section 1).

To send a file using the built-in Pactor-II command set one first establishes an ARQ link with the receiving station by sending "C callsign" to the local modem, where callsign applies to the called station. The log file records this action with the line

30.06.00 15:52:09 C DER1

Once the link is established (which is usually confirmed in a second or two by a

***CONNECTED

string from the local modem), the test program answers the BBS prompt with the appropriate "Send callsign…" command. For the built-in Pactor-II mailbox this is the callsign of the message recipient followed by the title of the message. (We send to our own callsign at the remote mailbox to give ourselves deleting permission in the mailbox) This "send-sequence" is recorded in the example above with the lines

30.06.00 15:52:20 BBS prompt rcvd
30.06.00 15:52:20 S BED1 file.25k

After transmission of the send command, the test programs calculate and record the size of the message to be sent, including the end-of-message (EOM) string:

msg file size = 25000 uploads = 3 last_upload = 6006

The number of "uploads" has to do with our method of flow control. To prevent overflow of the local (sending) modem's 12k transmitting buffer, we do a first upload of at most 10k characters followed by subsequent uploads (if the file was big enough to require it) of 9k characters each. (More sophisticated flow-control is possible in the PTC-II hostmode.) After each upload, subsequent uploads to the buffer are started only after a count of 9k "echoed" characters shows that all but 1k of the previous upload has been received and acknowledged by the remote station. (Activating the desired "remote-echo" mode is done by setting the modem to "Terminal Mode 1." Note that remote-echoed characters are not sent over the air; they are simply sent by the local modem to the local PC in response to acknowledgments from the remote station.) Leaving 1k in the buffer ensures that the modem always has characters to send until the message is gone.

In the example above the message file is 25k bytes long, so the first upload is 10000 byes, the second is 9k bytes and the third is 25k - (10k + 9k) + (EOM size) bytes. Since the EOM we send to the Pactor-II mailbox is \r*nnnn*\r, where \r is a linefeed, the third and last upload is 25000 – 19000 + 6 = 6006 characters long, as indicated.

After the last upload is finished (but before all the characters in the transmit buffer have been echoed), the software sends the EOM:

30.06.00 15:58:44 Sending EOM

Shortly after the beginning of the first upload the modem starts sending the contents of its TX buffer. The test software also begins counting remote-echoed characters for flow control, and recording “status bytes” sent over the serial port by the local modem.

Status bytes reflect link changeovers (CHOs) sent by the master (“master CHO”) or received from the slave (“slave CHO”). They also indicate when traffic has been sent by the master (“master traffic”), master/slave idle states, frame repeat requests from the slave (labeled “master request” since the corresponding status byte is recorded *by the master*) and erroneous frames received from the slave. The latter are labeled “master error” since the corresponding status byte is again recorded *by the master*. Request and error counts give a useful record of how hard the protocol worked to deliver messages. In the example above, the log file recorded

30.06.00 15:52:45 0xFFA4 slave CHO
30.06.00 15:52:45 0xFFAA master traffic
30.06.00 15:52:45 0xFFA9 master request
30.06.00 15:52:45 0xFFAA master traffic
…
30.06.00 15:55:30 rem read 1 10000 chars sent, 9000 rem-echoed chars rcvd
…
30.06.00 15:55:58 0xFFA9 master request
30.06.00 15:56:02 0xFFAA master traffic
30.06.00 15:56:06 0xFFA8 master error
…
30.06.00 15:58:36 rem read 2 19000 chars sent, 18000 rem-echoed chars rcvd
30.06.00 15:58:44 Sending EOM
…
30.06.00 16:00:07 rem read 3 25000 chars sent, 25000 rem-echoed chars retrieved
…

The actual status byte sent by the modem is written to the log file in hexadecimal format (e.g., 0xFFA9) followed by an explanatory label (e.g., “master request”). If repeat requests and erroneous frames from the message-receiving station are received (the former is quite common and the latter relatively rare) they are time-tagged, counted and logged as illustrated above.

Recall that the “master” label here means that the corresponding status was *logged by the master* (the local, message-sending station). A repeat request (“master request”) or received error (“master error”) is actually *sent or committed by the slave* (the remote, message-receiving station) and is only *recorded* by the master.

Following a successful message transfer (anticipated by a remote-echo count that equals the sent-message size), the software logs the number of the last remote-echoed-character reading session (here 3), the total number of message characters sent and the total number of remote-echoed chars retrieved (the last two numbers should be the same):

30.06.00 16:00:07 rem read 3 25000 chars sent, 25000 rem-echoed chars retrieved

As soon as the appropriate confirmation arrives from the remote station that it has stored the error-free message, we get what we’ve been waiting for: the statistics of the transfer:

30.06.00 16:00:10 Message "file.25k" filed.
LinkTime 2 s NegotiationTime 22 s
25006 bytes sent in 456 s; 54.8 bytes/s 35 TxARQs 1 TxERRs
30.06.00 16:00:10 BYE sent...

The link time is the time between the connection request and arrival of the ***CONNECTED string. This time is usually a second or two except in very poor conditions. The negotiation time is the time between establishment of the connection and the beginning of the message upload. This is the time taken up by sending, receiving and acknowledging the subject prompt, etc.

Next come the number of characters sent and the transfer time. The latter is the time between the start of the upload (right after the end of the negotiation phase) and confirmation from the message-receiver of correct message reception (here 456 seconds, or about seven-and-a-half minutes). The throughput in bytes per second (54.8 bytes/s) is the uncompressed message size (here 25006 bytes) divided by the transfer time.

The remaining statistics are the numbers of repeat requests and frame errors received by the master (message-transmitter). These are labeled TxARQs and TxERRs since they are recorded at the massage-transmitting station. Here there were 35 repeat requests and one error from the message recipient.

After logging the transfer data the program sends a disconnect request (“BYE sent”). Logging of a successful transfer is finished when the program receives the “***DISCONNECTED” prompt from the local modem:

30.06.00 16:00:21 Disconnected
Trial 1: Successful xfer to PTC-II MBOX

As noted above, we have found detailed transfer-log data such as this to be useful on many occasions in understanding protocol behavior and analyzing bugs in the testing software.

All of the files sent for this report consisted of readable text. Through experimentation we deduced that the “optimal” file size for throughput assessment of Pactor-II (close to highest throughput with smallest test time) was between 10 and 40k bytes, and most of our files had sizes in that range. This “optimal file-size range” for throughput assessment applies to most of the modern HF ARQ protocols we have evaluated.

4. Recorded Throughput-Data Format

The data archive file into which the results of each transfer were written by the test software contains the date-time group at the time (GMT) of the transfer, callsign of the receiving station, callsign of the sender, the mailbox system called (PTC=built-in Pactor-II or ArM=Airmail), the compression mode (0, 1 or 2), the frequency in MHz, the link time in seconds, the uncompressed message-file size in characters, the message transfer time in seconds, the throughput in characters/s, the negotiation time in seconds, the number of repeat requests and the number of bad frames from the file-receiver.

Here are excerpts from the NVIS transfer-data file for Pactor-II tests run in March and July 2000:

```
11.03.00 11:52:18 DER1   POR1  ArM 1    7.5xx  1 12010  306 39.2  32  12  7
11.03.00 12:02:10 DER1   POR1  ArM 1    7.5xx  2 20095  479 42.0  37  25  0
11.03.00 12:15:09 DER1   POR1  ArM 1    7.5xx  1 20093  430 46.7  23  30  0
11.03.00 14:12:01 BED1   POR1  PTC 2    3.1xx  1 20091  385 52.2  12   6  0
11.03.00 14:23:42 BED1   POR1  PTC 2    3.1xx  1 25006  460 54.4  12   3  0
20.07.00 02:45:49 KB1JY  KB1PZ ArM 0    3.615  1 20008  967 20.7  27  42  2
20.07.00 04:41:44 KB1JY  KB1PZ ArM 0    3.615  1 30010 1641 18.3  35 109  21
```

The first line, for example, records a 12010-character file sent at 11:52:18 GMT (6:52:18 AM local time) by POR1 in Portland, Me., to an Airmail mailbox at DER1 in Derry, N.H. The message was sent in compression mode 1. Linking took 1 second, negotiation 32 seconds and the transfer 306 seconds, for a throughput of 39.2 chars/s. There were 12 frame-repeat requests from Derry and 7 erroneous frames from Derry were received in Portland.

Files like this are opened and analyzed by a data-analysis program described in the next section.

5. The Data-analysis Software

The results in the data archive were analyzed off-line by a program called sum_ptc.c. This program reads the archive file line-by-line looking for various strings. As it moves through the file to the end-of-file indicator, the program keeps running totals of throughput and other data corresponding to the strings, from which it calculates statistics such as the average and standard deviation of the throughput. The statistics are written to a summary file after the pass through the archive file. Switches in the summary code are set before each run to pick out specific data (corresponding to various string combinations) for analysis. For example, we select lines with particular mode settings to pick out data sent in each compression mode, and use the date-time group to distinguish daytime from nighttime transfers. Since the summary program was written to analyze archive files of fixed format but arbitrary length, summaries of the data collected so far can be made at any time.

Shown below is the output of the summary program for all Pactor-II NVIS tests run from May 1999 to early February 2000 (a subset of all such data). For this output we set the software switches to compute throughput statistics for *files sent at night to a Pactor-II mailbox in compression mode 2*.

```
Statistics calculated from archive.ptc on 03.02.00 00:11:23
NIGHTTIME PTC-BBS COMPRESSED(2) PTC-II TRANSFERS
SAMPLE SIZE      = 56        E(SENT)           = 18730 bytes
E(TRANSFER TIME) = 493  s    sd(TRANSFER TIME) = 190   s
E(LINK TIME)     = 2    s    sd(LINK TIME)     = 1     s
E(NEGO TIME)     = 27   s    sd(NEGO TIME)     = 23    s    SAMPLE SIZE      = 48
E(THRUPUT)       = 40   cps  sd(THRUPUT)       = 17    cps  sd(mean_THRUPUT) = 2.3 cps
max(THRUPUT)     = 98   cps  Median(THRUPUT)   = 38    cps  E(TPUT/Hz)  = 0.080 cps/Hz
XFER FAILURES    = 1         P(XFER SUCCESS)   = 56/57 = 0.98
```

The output shows that the expected (average) throughput for 56 mode-2-compressed file transfers was about 40 characters per second (cps) and that the largest observed throughput in this mode in these conditions was 98 cps. The average sent-file size was 18,730 bytes. The sd(THRUPUT) reflects the spread of throughput measurements about their average. Roughly speaking, about two-thirds of a set of measurements will be within one standard deviation (here 17 cps) of their mean and over 90% will be within two standard deviations of their mean.

We also calculate the “standard deviation of the mean throughput” [sd(mean_THRUPUT)] in characters per second and the average throughput per Hertz of signaling bandwidth. The standard deviation of the mean throughput (equal to the standard deviation of the throughput divided by the square root of the sample size) is an assessment of the variability of the mean itself (which has its own statistical variability). The sd(mean) above suggests that our sample size in this case is big enough to make us confident that if we collected many more throughput measurements under roughly the same conditions, we would not get an average throughput that differed from the one above by more than about two characters per second.

To estimate the average throughput per Hertz [E(THRUPUT/Hz)], we divide the average throughput by the signaling bandwidth. For Pactor-II, the signaling bandwidth is 500 Hz. In this case the throughput per Hertz was 0.08 cps/Hz.

The average link, negotiation and transfer times were 2, 27 and 493 seconds. (We started measuring negotiation time a few days after the start of testing so the sample size of the average negotiation time is slightly smaller than the sample size of the throughput.)

The median throughput (the throughput above and below which half the samples lie) was 38 cps. The median can be used as a check on whether or not the mean is a good summary of performance. Since the median and mean are close, the mean is a realistic measure of performance in this case.

The last line of the statistical summary gives the percentage of successful file transfers. Unsuccessful transfers occur when, after a successful link, the number of times the modem tries to send a data frame exceeds a programmable limit (MAXERRS) causing the protocol to time out and terminate the link. (We set MAXERRS = 30.) As in our other testing, we do not include failures to link in our transfer success ratios. In the excerpt given above, 57 transfer attempts resulted in ARQ links, one of which was terminated when the link timed out before the message file got through. This led to a transfer success ratio of 56/57 or approximately 98%.

Figure 2 gives a scatter plot of all throughput measurements made to a Pactor-II mailbox through the beginning of January 2000 (only about half of the testing period). On the ordinate is throughput in characters/s and on the abscissa time-of-day (GMT). The plot gives a good picture of the spread of throughput, which is large, and typical of HF data communications. (Throughputs lie between about 10 and 110 cps.) One can discern in this plot the clear advantage of compression (Modes 1 and 2) over non-compression (Mode 0). Also discernable, perhaps, is the slight superiority of Mode-2 over Mode-1 compression.

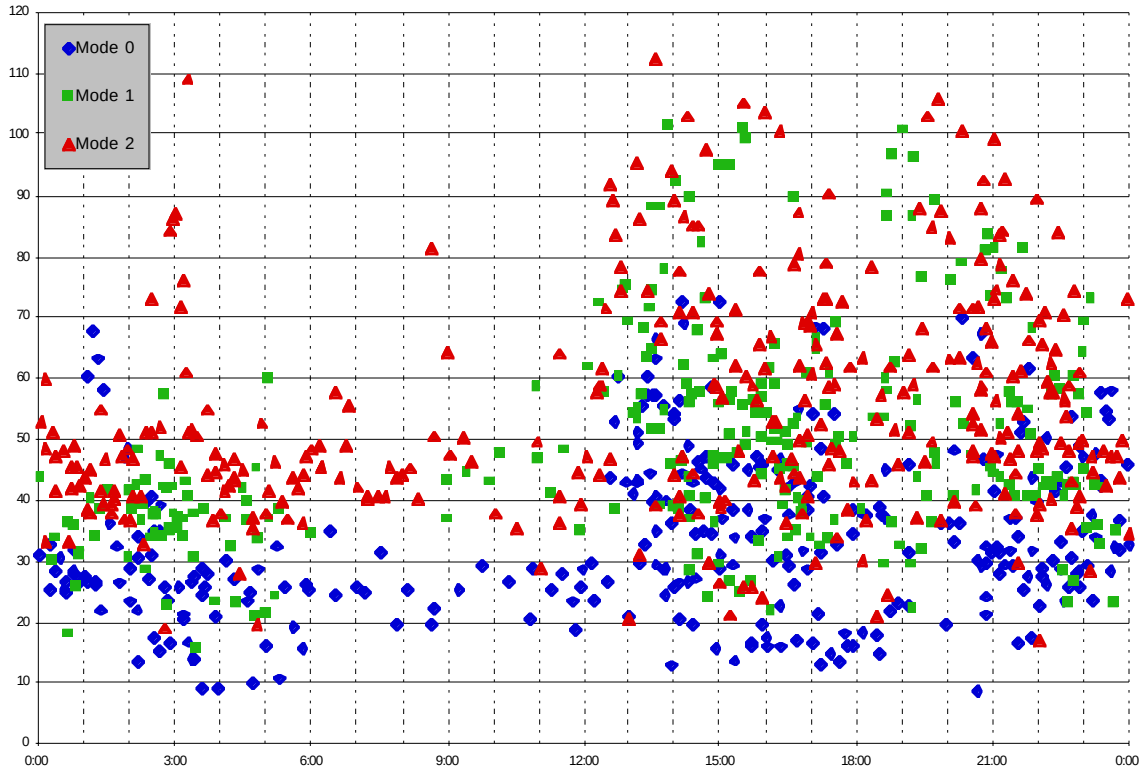

Fig. 2. Scatter Plot of Throughput to the Pactor-II Mailbox

6 . Statistical Summary of Throughput Results

The results of our NVIS tests of the Pactor-II system (as of July 2000) are summarized in Tables 1 through 6 below. They correspond to transfers to Pactor-II and Airmail mailboxes, in each case with compression modes 0, 1 or 2. The first column in each table gives the average throughput and its standard deviation, the average throughput per Hertz, the standard deviation of the mean throughput and the maximum observed throughput. The second column gives the number of transfers and the probability in percent of successful transfer [P(good xfer)] in each case. The third column gives the median throughput. The fourth column gives the mean of the link, negotiation and transfer times in seconds and the fifth column the average number of bytes in the original, uncompressed sent-message files.

Table 1. Pactor-II Mailbox Throughput Data in Compression Mode 0

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	Median Tput	Average Link Time Neg. Time Xfer Time	E(No_char)
Mode 0 Day	36 cps 15 cps 0.07 cps/Hz 0.9 cps 72 cps	288 97%	34 cps	2 s 18 s 608 s	18852
Mode 0 Night	34 cps 13 cps 0.07 cps/Hz 0.9 cps 87 cps	227 99%	30 cps	1 s 17 s 634 s	19963

Table 2. Pactor-II Mailbox Throughput Data in Compression Mode 1

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	Median Tput	Average Link Time Neg. Time Xfer Time	E(No_char)
Mode 1 Day	59 cps 24 cps 0.12 cps/Hz 1.5 cps 104 cps	245 99%	56 cps	1 s 16 s 518 s	23702
Mode 1 Night	45 cps 16 cps 0.09 cps/Hz 1.1 cps 102 cps	212 100%	42 cps	1 s 17 s 506 s	20735

Table 3. Pactor-II Mailbox Throughput Data in Compression Mode 2

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	Median Tput	Average Link Time Neg. Time Xfer Time	E(No_char)
Mode 2 Day	62 cps 23 cps 0.12 cps/Hz 1.2 cps 110 cps	359 99%	58 cps	1 s 16 s 445 s	23998
Mode 2 Night	47 cps 19 cps 0.09 cps/Hz 1.2 cps 109 cps	234 99.9%	44 cps	1 s 19 s 509 s	21522

Table 4. Airmail Mailbox Throughput Data in Compression Mode 0

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	Median Tput	Average Link Time Neg. Time Xfer Time	E(No_char)
Mode 0 Day	36 cps 14 cps 0.07 cps/Hz 1.0 cps 73 cps	210 97%	35 cps	1 s 23 s 626 s	19858
Mode 0 Night	30 cps 11 cps 0.06 cps/Hz 0.9 cps 68 cps	152 95%	28 cps	1 s 23 s 783 s	20480

Table 5. Airmail Mailbox Throughput Data in Compression Mode 1

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	Median Tput	Average Link Time Neg. Time Xfer Time	E(No_char)
Mode 1 Day	53 cps 19 cps 0.11 cps/Hz 1.1 cps 102 cps	270 98%	49 cps	1 s 23 s 474 s	22467
Mode 1 Night	41 cps 11 cps 0.08 cps/Hz 0.9 cps 74 cps	162 98%	39 cps	1 s 25 s 566 s	21502

Table 6. Airmail Mailbox Throughput Data in Compression Mode 2

File Type & Time	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No. Xfers P(good xfer)	Median Tput	Average Link Time Neg. Time Xfer Time	E(No_char)
Mode 2 Day	57 cps 20 cps 0.11 cps/Hz 1.3 cps 112 cps	244 98%	54 cps	1 s 22 s 446 s	22425
Mode 2 Night	48 cps 13 cps 0.10 cps/Hz 1.1 cps 109 cps	144 98%	46 cps	1 s 23 s 495 s	22622

7. Discussion of Pactor-II Performance Statistics

Tables 1 and 4 (transfers to Pactor-II and Airmail mailboxes in compression mode 0) show that the inherent (no compression used) over-the-air average throughput of Pactor-II on our NVIS links is around 36 characters per second (averaging the two mailbox cases). This is equivalent to an average delivery time of about ten minutes for a 20k text file. The nighttime NVIS throughput average for uncompressed files is about 32 cps. The standard deviations of these throughputs are around 15 cps in daytime and 20 cps at night. These imply that there is considerable variability in performance even with the same mode (see Fig. 2). This is typical of HF data transfers. The main cause of the lower nighttime throughput is probably QRM. Standard deviations of the means are about a character per second. This leads to high confidence that we have large enough sample sizes for our means to characterize real NVIS throughputs.

Figure 3 gives a plot of throughput vs. time of day for sixty-one 30k-files sent with Mode 2 compression half-an-hour apart. (The time between the end of one transfer and the beginning of the next was half-an-hour.) The transfers occurred over the course of 42 hours in early July 2000, when the ionosphere was relatively undisturbed. The link was from Derry, N.H., to an Airmail mailbox in Bedford, Mass. (30 miles away). The plot illustrates both diurnal and statistical variability of performance over NVIS channels. Note, in particular, the rapid rise in throughput at sunrise as the MUF increases, and the generally lower throughput at midday, which was caused by D-layer absorption.

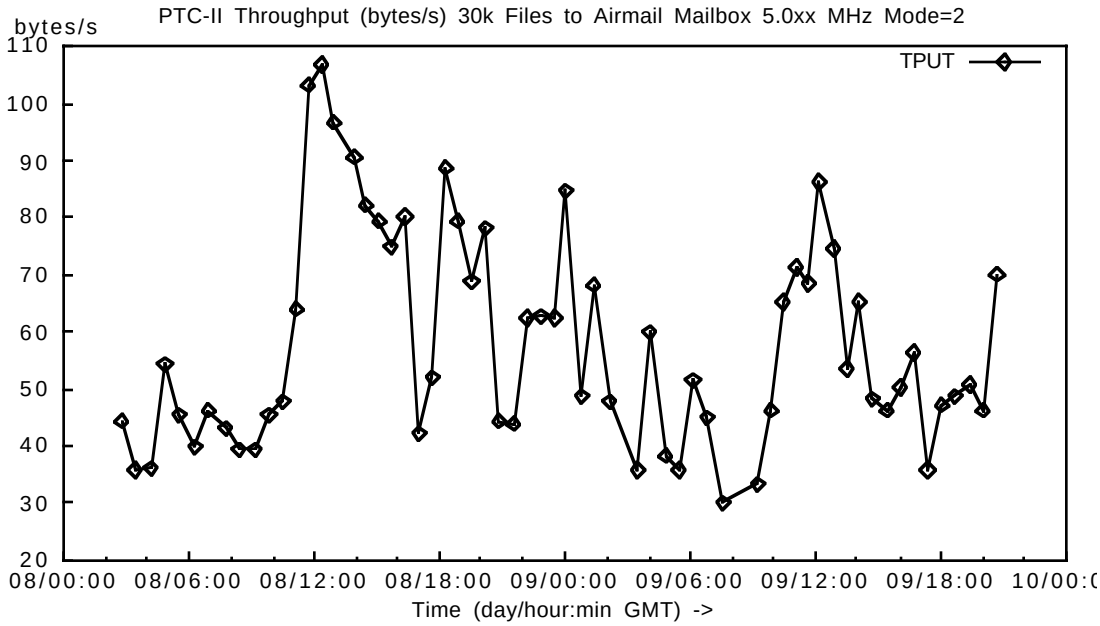

Figure 3. Throughputs for 61 Files Sent Half-an-Hour Apart

Measured maximum throughputs are generally large—at or near the theoretical maxima. These are not very useful statistics, however, since these maxima are rarely attained.

The probability of correct message reception was above 95% in all cases. This is also typical of well-designed HF ARQ protocols with their numbers of retries set

conservatively. (Ours were generally set at 30.) This says that if Pactor-II can establish a link, it can almost always (even at night) complete a file transfer. (The transfer success rate given a link can be raised to almost 100% by raising the retry limit, at the expense of wear and tear on radios and amplifiers.)

Although we were usually successful in choosing frequencies that supported linking, a number of link attempts (perhaps 10% of them) failed when there was too much QRM on all available frequencies or when the MUF dropped below our set of available operating frequencies. An automatic link establishment (ALE) system that uses sounding data to choose frequencies for link attempts would probably have reduced this number if a large enough set of linking frequencies was available. Linking statistics are not included in our data: all of our transfer-failure percentages are conditioned on a link's having been established before each transfer attempt.

The Pactor-II and Airmail systems (the first in the modem, the second installed on Windows PCs) do not appear to have significantly different throughputs associated with them. For that reason we have combined their statistics in the following discussion.

Average throughputs for text files compressed in Mode 1 are about 55 cps during the day, and 43 cps at night (Tables 2 and 5). Mode 1 compression led to about 50% higher throughput than using no compression.

Mode 2 compression produced day- and nighttime throughputs of about 60 and 48 cps. This mode produced the best performance, probably owing to the wider choice of compression methods made by the protocol in that mode.

It should be noted that how much a compression method reduces the size of a file, and whether it even reduces it at all, depends on the file's type. Our results apply only to text files.

8. Throughput Comparison with Clover II and Certain Military Protocols

For some time newsgroup and listserver participants have been debating whether Pactor-II has higher throughput than HAL Corporation's Clover II system. Some of the claims in this debate seem to rest on anecdotal or otherwise questionable evidence. The debate is perhaps further clouded by the fact that HAL sells other Clover systems (the P38 and Clover2000) that generally have lower (P38) and higher (Clover2000) performance than Pactor-II. Since we have also carried out extensive on-air NVIS measurements of the Clover II system (as well as of the P38 and Clover2000 systems—see the references), results from our measurement campaigns may push the Clover II vs. Pactor-II debate a little closer to resolution.

Both Pactor-II and Clover II operate in 500-Hz bandwidths, use powerful forward error correction, have waveforms that are either PSK or fairly similar to it and employ sophisticated ARQ schemes that rapidly adapt modulation modes and FEC coding rates to channel conditions. Both systems have had their adaptation algorithms tweaked and tested over the air numerous times during several years with resultant firmware upgrades. These facts might cause one to be skeptical of claims that one system has much higher throughput than the other.

Before we compare the two systems' performance on NVIS links we should list the similarities and differences in conditions under which we tested each system.

Similarities

Links, transmitter output powers, antennas and average sent text-file sizes.

Differences

Transceivers, sunspot numbers, freedom to set "bias" for the Clover II protocol and scheduling of nighttime message transfers.

The differences deserve elaboration. Two of the four transceivers we used for some of the Pactor-II tests were Yaesu FT–1000MPs with 500-Hz SSB filters. These filters may have improved performance somewhat for the Pactor-II tests. However, since the frequencies we used for transfers to the Pactor-II and Airmail mailboxes were seldom bothered by QRM, this difference probably had little effect on results.

Our Clover II tests took place over six months between September 1997 and February 1998, when the sunspot number was about half what it was for the more recent Pactor-II tests. Although this didn't make much difference in the set of frequencies we chose for NVIS testing (mostly 3-5 MHz in both cases), there was more flare activity during the end of the Pactor-II testing period than during the Clover II testing. However, since scheduling conflicts kept us from testing during most flare aftermaths, sunspot number differences had little effect on our throughput data.

The Clover protocol allows users to set a parameter called *bias* that determines how aggressive the protocol should act in various perceived channel conditions. The biases, which are associated with frame lengths and the FEC coding rate, are Fast, Normal and Robust, with Fast appropriate when the channel is good and Robust when it's bad. We adjusted the bias during our Clover measurements. Although this added additional variance to our Clover results, comparisons of achieved throughput after most bias changes suggested that our changes usually raised or at least maintained throughput. We have concluded that Clover bias adjustments (viewed as an integral part of the protocol) did not distort our throughput comparison.

The differences in nighttime scheduling had to do with the ease of writing code to schedule tests with the Pactor-II and Clover II modems. Pactor-II has a relatively simple set of text-based commands and we were able to write testing programs for it relatively quickly. Clover II, on the other hand, has a rich and fairly complicated control language and we didn't have time to write and debug scheduling code with it. This forced us to run our nighttime Clover II tests in the *evening* (when we were awake) rather than spread them more evenly over the whole night. The effect of this is hard to assess accurately. On the one hand, rarely operating Clover II in the middle of the night avoided occasional operation above the MUF then, which may have lowered Pactor-II throughput on occasion. On the other hand, some NVIS frequencies are more crowded in the evening than in the middle of the night, which may have worked to the detriment of Clover II. We decided that these differences were murky enough to preclude a useful comparison of nighttime performance with our data.

A comparison of the daytime performance of the two protocols is given in Table 7. (Daytime was defined for both sets of data as 5 AM to 5 PM local time.) The table gives throughput and other statistics for compressed and uncompressed text file transfers. In the case of Pactor-II we chose Mode-2 data for comparison in the compression case, since Mode 2 produced the highest Pactor-II throughput. (Clover II uses a fixed, dictionary-based compression technique from the PKLib package.)

Our long-term NVIS measurements show Pactor-II with slightly higher daytime throughput than Clover II for text files sent in compressed or uncompressed modes. The differences in transceivers used and the fact that the tests were separated by more than a year are probably enough to explain the differences in throughput. The two systems had almost the same high probability of successful message delivery. Our data suggest that users wanting to choose between the Pactor-II and Clover II systems should use other criteria than throughput or message-delivery probability.

Table 7. Pactor-II vs. Clover II Performance

File Type & Time	E(thruput) Chars/s	Number Of Transfers	Maximum Thruput Chars/s	P(success) Percent	E(No_char)
Uncompr. Day					
Pactor-II*	36	288	72	97	18852
Clover II	34	239	69	98	31346
Compr. Day					
Pactor-II*	62	359	110	99	23998
Clover II	52	172	128	98	33986

* Pactor-II: Mode 2 to PTC2 MB, as of 30 June 2000

The results in Table 7 may be compared with those for the US Federal Standard 1052 and NATO STANAG 5066 protocols, which use a much wider bandwidth (usually 3 KHz) and much more expensive modems ($3k and up). Modern military modems usually have data-directed equalizers that deal very effectively with multipath. The currently fielded versions can operate at up to 2400 bits per second and adaptive military protocols often achieve that rate even on NVIS links.

Our measurements with the FS-1052 protocol implementation by the Harris Corporation have produced uncompressed average daytime throughput of about 125 cps for 20-to-30k text messages sent over a 30-mile NVIS link. This is almost four times what Pactor-II and Clover II achieve on average. (We are not aware of any large-scale on-air measurements made yet with the STANAG 5066 protocol, but we expect it to have throughput similar to that of FS-1052.)

9. Concluding Remarks

We hope that our data will aid understanding of Pactor-II file transfers over HF and perhaps serve as a useful introduction to how Pactor-II works. The system is now employed all over the world by amateurs (especially for sending E-mail to and from boats and recreational vehicles). A number of international aid and other communications-providing organizations also use Pactor-II to send information across parts of Africa, where alternative means of long-haul communication are not available, or too expensive. Our data may shed light on why this effective, amateur-developed modem and its associated protocols are so widely used.

Acknowledgments

We are grateful to Jim Corenman (KE6RK) and Rick Muething (KN6KB) for advice on Airmail and Winlink. Jim and Rick also gave us essential help understanding flow control and buffering in the Pactor-II modem.

References

1. Ken Wickwire (KB1JY) et al., "On-air Measurements of HF TOR and Packet Throughput, Part I: Near-Vertical-Incidence-Skywave Paths," *Digital Journal*, March 1996.

2. Ken Wickwire (KB1JY), "On-air Measurements of HF TOR and Packet Throughput, Part II: One-hop Skywave Paths," *QEX*, June 1996.

3. Ken Wickwire (KB1JY), "On-air Measurements of HF Data Throughput: Results and Reflections," *Proc. 15th ARRL/TAPR Digital Communications Conference*, ARRL, 1996.

4. Ken Wickwire, "On-air Measurements of MIL-STD-188-141A ALE Data Text Message Throughput over Short Links," MITRE Internal Report, 1996. (Preview in *Proc. 15th ARRL/TAPR Digital Communications Conference*, ARRL, 1996.)

5. Ken Wickwire (KB1JY) et al., "On-air Measurements of Clover P38 Throughput," *Proc. 16th ARRL/TAPR Digital Communications Conference*, ARRL, 1997.

6. Ken Wickwire (KB1JY) et al., "On-air Measurements of Clover II and Clover 2000 Throughput," *Proc. 17th ARRL/TAPR Digital Communications Conference*, ARRL, 1998.

Email: kwick@mitre.org. Packet radio: KB1JY@W1ON.