On-air Measurements of HF Data Throughput
Results and Reflections

Ken Wickwire (KB1JY)
kwick@mitre.org
KB1JY@WA1PHY

1. Introduction

For the past year a team of colleagues and I[1] have been collecting and analyzing data on the throughput and other characteristics of various ARQ protocols available to hams and commercial users for HF work. This activity was motivated by discussions (especially among hams) about the relative merits of the new HF digital modes, such as PacTOR, GTOR, CLOVER II, CLOVER-2000 and PacTOR IT. Since the discussions often centered on throughput in various conditions, and we were already running several of the protocols, we decided to see for ourselves. This paper describes our assessment approach and measurement campaign, gives a summary of our main conclusions, and lists some findings worth noting before protocol choices are made and protocol performance is compared. The paper treats the packet and TOR modes in detail. More extensive reports on CLOVER II, NOS TCP/IP and the ALE orderwire will appear elsewhere.

2. Our Approach to Throughput Measurement

The randomly varying HF "channel"; that is, the combination of propagation conditions (fading, dominant ionospheric layer, etc.), and propagated and local noise, is generally agreed to be the worst radio channel. Over the past 20 years, powerful DSP techniques have been developed to tame this wild conduit and put it to work for data transmission, even when it resists being used for voice traffic. These techniques are now embodied in a surprisingly large and growing number of data transmission protocols whose performance is often impressive by HF standards. What people mean when they say (or write in an advertisement) that one of these protocols is better than another is not always clear, however.[2]

HF data transmission protocols can be divided for general discussion into two categories: those with automatic repeat request (ARQ) and those without. In some cases, the latter are called forward error correction (FEC) protocols, because they use FEC but not ARQ to control errors. ARQ protocols, which are almost always combined with FEC in modern systems, generally deliver error-free data, although there is no guarantee that the data will be delivered quickly. Since many users (especially military and governmental users, and operators of forwarding stations) demand error-free transmission, ARQ protocols have come to dominate technical discussions of late. For ARQ protocols, the definition of throughput, for example, is relatively straightforward; for protocols without ARQ, which can deliver erroneous data, the concept of HF throughput is more difficult to define. For these reasons we have decided to concentrate on ARQ protocols in our assessments.

There are three basic ways to assess the throughput (and most other kinds of performance) of an HF data transmission protocol. First, you can try to devise a

---

[1] See the Acknowledgments and reference list at the end of the paper.
[2] Although we recognize this shortcoming, we sometimes suffer from it ourselves.

mathematical model of what happens during data transmission with it, convert the model, if necessary, into computer code, and run the code (or your pencil) to assess (i.e., predict) performance. While this is sometimes advocated as the best approach by those too lazy, poor or otherwise constrained to try other approaches, it frequently produces unconvincing or incomprehensible results. Many believe that the modeling approach is best suited to the design stage of a new protocol rather than to the performance assessment stage.

Second, you can connect a pair of working systems through an analog or digital HF channel simulator, which can be set up to accurately produce various levels of some of the main phenomena of the HF channel, like multipath spread, fading and noise (usually Gaussian). Using a channel simulator with real hardware and software produces statistically repeatable results and allows reasonable-if not necessarily convincing— comparisons of different systems operating in so-called "standard channels," namely the ones whose statistics are programmed into the channel simulator. A channel simulator cannot, however, reproduce the statistical variations in transmission quality that occur on a real HF channel; it can't faithfully reproduce those caused by non-Gaussian (e.g., impulse) noise, intermittent and random interference by man-made signals with various waveforms, day-night transitions, and polar and equatorial propagation anomalies.

The third approach is through on-air measurements. This has the advantage that any one measurement is in a sense completely realistic and convincing, but the disadvantage that the conditions in which the measurement was taken are not generally repeatable. This means that producing *statistically* convincing assessments with this approach requires that a large number of measurements be made (resulting in a large sample-size) and that attention be paid to realistic and representative path lengths, power levels, antennas, diurnal variations and the spreads (variances) of performance statistics. This takes time and a lot of cooperation from several outlying stations.

We believe that a combination of channel simulator and on-air measurements leads to the most convincing assessment of ARQ performance in the HF bands.[3] The simulator creates repeatable channel extremes, while properly conducted on-air measurements comprise channel conditions the simulator hasn't been (or can't be) set for. This paper discusses a measurement campaign we've pursued in that belief for the past year. We should note that although our results allow an informative comparison of the throughputs of the protocols we've treated, the past year's measurements need to be continued to cover all seasons with all protocols and a wider range of sunspot numbers.

For our on-air measurements we try to write software that allows tests to be run automatically, so that the mistakes that we all tend to make during manual time-recording and data-entry can be avoided. (Sometimes—as with CLOVER and NOS TCP/IP implementations,—protocols come with their own interface software, and we use the existing software capabilities: That leads to some manual data logging.) So far, our software has been written in C for the Macintosh operating system, but it would work (with different I/O calls) on different operating systems.

With our software, we always measure file transfer time from the start of character-by-character uploading of a file to the sending modem to the time that a "message saved" (or equivalent) sent by the receiving station arrives via the sending modem to the test

[3] Of course, this is only true when the two approaches produce results that agree, at least qualitatively. For some simulator results that agree qualitatively with our measurements, see Refs. 1, 2 and 3.

program[4]. Waiting for a "message saved" makes the transfer times a little longer than they would be if a human operator (or program) at the receiving station recorded the error-free arrival of the file. That in turn makes the throughputs slightly lower (i.e., more conservative) than they would be otherwise, but that is a small and reasonable price to pay for measurements that require only one program and no operator intervention during tests.

In the next section we'll describe the philosophy behind our choice of equipment and file types for most of the measurements made during our campaign.

3. Our Philosophy for Choosing Equipment to Use and Files to Be Sent

In choosing equipment (e.g., the KAM for PacTOR assessment) for our tests, we have been motivated not only by expediency (we have KAMs), but by the view that assessments of most interest to the most (prospective) operators are those of "common" operating setups; that is, ones widely available at competitive prices, and ones that offer a wide choice of operating modes and good technical support. While it is no doubt true that some implementations of PacTOR, for example, may have higher throughput than others because they use A/D quantization of bit energy or more advanced filtering, they are probably not in wide enough use to be part of a "common" operating setup. Nevertheless, if we had the time and money to buy and test all possible pairings of implementations of a particular protocol, we would gladly do it, since performance of the "best" or the "official" implementation is obviously of interest. In the meantime, we unselfishly invite others to fill in the gaps left by our work.

Likewise, we have chosen at this stage ASCII English text files of various sizes to compare the transfer capabilities of protocols. With due respect to the many who probably send text files written in other languages, we believe that sending such files represents a "common" application of the HF ARQ protocols described below. It should be borne in mind that languages other than English and German, and files with a non-standard distribution of characters (e.g., all upper-case characters), may benefit very little from the Huffman text compression used in current PacTOR implementations.

When a protocol like PacTOR, GTOR or CLOVER 11 comes with defaults for some of its "protocol-tuning" parameters (e.g., GTTRIES and GTUP for GTOR and BIAS for CLOVER II), we have used these defaults. This has been based on the belief that a common setup would not have these parameters changed. (Optimal tuning of such parameters is an area that should be looked at, however, and a few operators have recently started to do so.) For packet, on the other hand, we consider good values of PACLEN and MAXFRAMES to be highly dependent on channel conditions, and we juggled these values frequently to increase throughput in our tests (see below).

Finally, our philosophy says that if a protocol or common implementation offers data compression, then it should be used (if there's a choice) unless we think it might seriously expand a file (see the section below on data compression). This means that in the case of PacTOR we used Huffman compression and in the case of CLOVER II, we used the "PKLIB" compression (probably a Liv-Zempel-Welch variant) offered by the standard (i.e., "common") P38 terminal software provided by HAL with the modem.

---

[4] That is, we don't include linking and "negotiation" times in our throughput calculations. Others may view these times as legitimate components of transfer times.

In the next section we'll discuss some of the protocols we have assessed in the past year of our campaign along with some more advanced ones we may get to later.

4. ARQ Protocols Developed for HF Use and Their Throughput

Table 1 at the end of the paper lists most of the ARQ protocols that are in common use on the air[5]. With the exception of the last three, all are used in amateur work, and the last three, developed originally for military use, will probably enter the amateur world in some form in the next few years.

The table classifies each protocol according to its modulation scheme, signaling bandwidth, forward error correction capability, ARQ scheme, channel rate, character format, compression capability and measured throughput for "standard" (mainly lower-case) English text files. The throughputs with a O-symbol beside them have been measured as part of our campaign, with enough samples for statistical significance in current sunspot conditions. The other throughputs are based on channel simulator measurements. The measured throughputs for packet and the TOR modes are from an aggregate of short near-vertical-incidence skywave (NVIS) and longer one-hop skywave (OHS) paths. For CLOVER II and the ALE engineering orderwire, the throughputs are from only NVIS paths. (We expect to begin OHS tests with CLOVER II this summer, and to publish ALE, NOS TCP/IP and CLOVER results this year.)

It should be kept in mind that in agreement with our measurement philosophy, for our packet and TOR throughput measurements we have used Kantronics KAMs with firmware version 7.1 or higher. Other PacTOR implementations than the KAM's may yield higher or lower throughputs than ours. Note also that we have used the HAL P38 for all of our CLOVER II measurements; more expensive models, like the PCI-4000, have the computing power to select a 16-symbol signaling set, and may produce higher throughputs.

5. Differences Between NVIS and OHS Throughput for TOR and Packet

NVIS throughput is generally lower than throughput over "standard" one-hop skywave (OHS) paths; that is, fairly long paths on which fading (and resulting inter-symbol interference) is relatively slight, and average signal-to-noise ratios are comparatively high. In fact, one-hop skywave measurements paint a relatively optimistic picture of what operators can expect in day-to-day communications over HF.

However, some protocols appear to improve more than others when you go from NVIS to OHS operations. Tables 2 and 3 below (reprinted from recent papers listed in the References) give NVIS and OHS throughput and other statistics for AMTOR, PacTOR, GTOR and packet. (Recall that for packet, we juggled PACLEN and MAXFRAMES to increase throughput.)

Throughputs in the tables are in characters/sec and times are in seconds. The first column 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 links and the mean and standard deviation of the

---

[5] A recent newsgroup FAQ on signalling formats lists a number of ARQ protocols in use in Europe, the CIS and Asia that we never heard of, so we may be misleading our readers with this statement. Most of these protocols may be rather old and inefficient, like AMTOR, but we can't be sure.

"link time." The third column gives the number of "negotiation times" and the mean and standard deviation of the negotiation time. Link time is the time (in seconds) between sending the link command and receipt by the program of the "LINKED TO" notification. Negotiation time is the time between sending the link command and the start of message-file transfer. In most cases there are fewer negotiation than link times because we started measuring the former part way through the campaign. The fourth and fifth columns give the means and standard deviations of the transfer time and the number of transferred characters.

The standard deviation of the mean (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`'s in the tables suggest that our sample sizes are big enough to give us pretty high confidence that if we collected many more throughput measurements under roughly the same conditions, we would not get average throughputs that differed from the ones above by more than about a character per second.

To calculate the average throughputs per Hertz [`E( tput /Hz )`], we divided the average throughput by the average signaling bandwidth. We calculated the latter using the formula for "necessary telegraphy bandwidth" (from the 1992 Dept. of Commerce *RF Management Handbook)* BW = *baud rate* + 1.2 x *shift,* where *shift* for most of our TOR and packet tests was 200 Hz. For AMTOR, the baud rate is of course 100; for PacTOR, GTOR and packet, we used the rough average of the baud rates chosen automatically in the PacTOR and GTOR modes and manually in packet. Our estimates of these average baud rates were 150 (PacTOR), 200 (GTOR) and 200/300 (NVIS/OHS packet). The resulting average bandwidths were AMTOR: 340 Hz, PacTOR: 390 Hz, GTOR: 440 Hz and NVIS/OHS packet: 440/540 Hz.

The majority of NVIS measurements were at 3.606 MHz LSB, with some at 7.085 MHz LSB and 1.815 MHz LSB. They were made during the winter over all daylight hours and also in the evening, after dark; a few were made in the middle of the night. Interference usually prevented throughput tests from about six to ten in the evening (2300Z-0300Z) on 3.606 and 7.085 MHz.

Most of the OHS measurements were at 10.141 MHz LSB. About 20% were taken at 3.640 MHz, 14.075 MHz, 14.123 MHz or 18.075 MHz, all LSB. These measurements were made during the winter and spring over all daylight hours and also in the evening, after dark. However, interference often prevented throughput tests from about six to ten in the evening (2300Z-0300Z) on 3.640 MHz. The NVIS and OHS tests covered roughly the six-month period from November, 1995, to April, 1996.

All measurements were made using transmitter output of around 100 watts, and all stations generally used sloping longwires or dipoles. These setups can be viewed as embodying average station capabilities. NVIS paths (in New England) were from 30 to 200 miles long and OHS paths (on the east coast and from New England to the midwest) were from 400 to 1200 miles long.

In discussing the TOR and packet results let's start with some observations on NVIS and OHS communications quality in general. First of all, note that we haven't collected data on the fraction of tries in each mode that we were successful in linking, "negotiating" and transferring a file. However, we have found that over OHS paths, the three TOR modes and packet can get files through in the absence of strong interference on most tries during the day. This is in contrast with our NVIS results, which showed that except during the mid-morning and mid-afternoon "windows," packet and AMTOR had transfer probabilities well below one.

Under difficult conditions (especially those on NVIS paths leading to marginal SNRs) PacTOR occasionally out-performed GTOR in terms of throughput, although GTOR has higher average throughput. This seems to confirm the rumor that GTOR needs high SNRs for high performance. However, this "role-reversal" happened much less frequently over OHS than over NVIS paths.

In the early evening on both NVIS and OHS paths, there was sometimes increased interference on the frequencies we used. During these periods of interference it was rare to see a file transferred. (An automatic link establishment (ALE) system, such as prescribed in MIL-STD- 188- 141 A, could probably have found a frequency without interference.)

Table 2. Statistical Summary of NVIS Throughput Data

Mode	E(thruput) sd(thruput) E(tput/Hz) sd_mn(tput) max_tput	No_links E(lnk_tm) sd(l_tm)	No_neg_tms E(neg_tm) sd(neg_tm)	E(xfer_tm) sd(xfer_tm)	E(No_char) sd(No_chr)
AMTOR	5.20 cps 1.13 cps 0.015 cps/Hz 0.08 cps 6.33 cps	226 3.02 s 3.16 s	70 82.4 s 30.1 s	473.5 s 234.0 s	2358.1 974.7
PacTOR	17.83 cps 5.50 cps 0.046 cps/Hz 0.30 cps 25.10 cps	344 5.44 s 8.39 s	95 38.7 s 22.7 s	146.1 s 90.0 s	2452.7 1110.1
GTOR	23.52 cps 10.06 cps 0.053 cps/Hz 0.55 cps 44.12 cps	335 5.54 s 10.30 s	76 58.7 s 30.9 s	120.0 s 95.8 s	253 1.7 1580.3
packet	5.68 cps 3.53 cps 0.0 14 cps/Hz 0.25 cps 17.34 cps	197 8.73 s 10.48 s	119 102.7 s 66.9 s	556.7 s 367.6 s	2484.9 1043.1

Turning to particulars, you can see that AMTOR and PacTOR average throughputs don't differ much on OHS and NVIS paths, although there is a slight tendency toward greater statistical variation (as measured by standard deviations) on NVIS paths. This similarity of average throughputs may explain why you don't hear much about differences between performance on long and short paths in these two modes.

The big story is the differences between GTOR and packet performance on long and short paths. Average GTOR throughput on OHS paths was almost 50% higher on OHS paths than on NVIS ones (32 char/s vs 23 char/s). This may reflect the presence of consistently higher signal-to-noise ratios (SNRs) on OHS paths, since GTOR is said to thrive on high SNRs and suffer more than the other modes on low ones.

Packet throughput was two-and-a-half times higher on long paths than on NVIS ones (16 char/s vs 6 char/s). Some of this difference may have been caused by the fact that we restricted all our NVIS tests with packet to 200-baud operation. Although we based this restriction on observations of performance, it's possible that a more aggressive choice of baud rate on packet during the mid-morning and mid-afternoon "NVIS windows" could have raised NVIS packet throughput somewhat. However, this does not explain all of the improved performance, whose source must be the better OHS channel (fewer packet bit errors) .

Another striking difference appeared in the average packet negotiation times (OHS: 35 s, NVIS: 103 s). (Recall that negotiation time is the difference between the time a connection request is sent and the time file transfer starts.) This difference in average negotiation times apparently reflects the fact that the negotiation process for a packet BBS upload, which involves transmission of frames of various sizes, exposes packets at 206 baud much higher bit-error rates on NVIS paths than 300-baud negotiations over OHS paths.

Table 3. Statistical Summary of OHS Throughput Data

Mode	E(thruput) sd(thruput) E(tput/Hz) sd_ mn(tput) max tput	No_links E(lnk_tm) sd(l_tm)	No_neg_tms E(neg_tm) sd(neg_tm)	E(xfer_tm) sd(xfer tm)	E(No_char) sd(No chr)
AMTOR	*5.70* cps 0.80 cps 0.017 cps/Hz 0.08 cps 6.33 cps	104 2.62 s 3.8 1 s	*92* 69.7 s 15.2 s	*543.2* s 109.8 s	*3009.6* 98.1
PacTOR	20.19 cps 5.49 cps 0.052 cps/Hz 0.44 cps 25.00 cps	153 *4.70* s 6.53 s	139 40.8 s 29.4 s	176.1 s 105.2 s	3058.8 308.5
GTOR	32.30 cps 9.88 cps 0.073 cps/Hz 0.79 cps 44.12 cps	158 4.44 s 7.96 s	144 50.9 s 21.7 s	119.9 s 102.6 s	3126.6 501.4
packet	15.67 cps 4.58 cps 0.029 cps/Hz 0.44 cps 24.59 cps	108 6.46 s 8.50 s	108 34.6 s 17.7 s	221.9 s 141.4 s	2975.0 259.8

Maximum observable TOR throughputs were about the same for NVIS and OHS paths, although, as mentioned above, individual measurements came closer to their maxima more often on the long then on the short paths. On packet, we achieved maximum OHS throughput of about 25 char/s vs about 17 char/s for NVIS.

6. Discussion of Packet Results

Our packet experiments over both NVIS and OHS paths have led to surprising results in view of what we have read on newsgroup discussions and elsewhere. For example, over OHS links we have consistently achieved average packet throughputs two to three times higher than average AMTOR throughputs, although not quite as high as PacTOR, and only about half of the GTOR average (see Table 3 above). The parameters we have adjusted to do this are PACLEN, MAXFRAMES, .FRACK, SLOTTIME, RESPTIME and PERSIST, and we have done all our OHS file transfers at 300 baud. (Since we have tried to choose frequencies and times where there is little interference, we have set PERSIST very high and FRACK, SLOTTIME and RESPTIME low for aggressive use of the channel.)

These high packet throughputs have been achieved, however, *only during the day,* and by means of very frequent, manual, changes of PACLEN and MAXFRAMES. Furthermore, we have managed to find frequencies that were by and large free of significant interference from other signals (this appears to rule out most of the 20m band). For example, we have often been able to transfer files over both NVIS and OHS paths with combinations like PACLEN = 100 and MAXFRAMES = 5 in the absence of contention, which may be a revelation to some hams who have tried HF packet.

As a general rule, as packet begins to work in the morning on our links, values of PACLEN/MAXFRAMES around 40/1 work best. From mid-morning till late afternoon, combinations like 80-100/4-7 often lead to high throughput. As the bands begin to deteriorate, it's back to near 40/1. PACLENs greater than about 120 bytes usually suffer too many bit errors on our NVTS and OHS links to be worth trying.

After about 5 PM local time during the winter, throughput rapidly falls, and for most of the evening, getting files through in any mode was difficult. (We got some NVIS transfers through in the middle of the night during the winter, but we didn't try any OHS transfers in the middle of the night.) On our links, trying a lower ham frequency in the evening usually led to increased interference, against which none of the modes did a great job.

Our experience with HF packet on OHS and NVIS links has convinced us that an adaptive protocol that adjusted HBAUD, PACLEN and MAXFRAMES using feedback on throughput could go a long way toward polishing HF packet's tarnished reputation. However, with much better systems now available at reasonable prices, it is probably no longer worth developing such a protocol.

7. The Effects on Throughput of Data Compression and File Type

Three of the protocols we have assessed over the air provide one or more types of optional or hard-coded data compression: PacTOR has (optional) Huffman compression, GTOR has hard-coded Huffman and run-length compression and CLOVER II with the HAL interface software has one or more hard-coded compression techniques from the so-called "PKLIB" suite. Other and future protocols may also include one or more compression capabilities[6]. Of course, even when a protocol doesn't include compression, the user is free to compress his files off-line before he sends them, provided that the protocol can handle the compressed format and the receiving station can de-compress the files. (For an introduction to data compression see Ref. 7.) As mentioned above, we almost always choose the Huffman option in PacTOR transfers of English text files.

How much a file gets reduced by a compression technique is strongly affected by the file's type and the technique's approach, so that the user must have some understanding of the interplay of the two if he wants to use compression for high throughput.

In general, the closer the distribution of a file's ASCII characters to the distribution of characters in "typical" English (or other language to which the Huffman code has been tailored) text, the more Huffman will compress the file. The more repeated contiguous characters or bytes (e.g., spaces) in the file, the more run-length coding will compress it. The more repetitions of byte-pairs in a file ("an," "th" in "the," etc.), the more so-called Markov coding (multi-level Huffman) will compress it. The more repeated byte strings in the file, the more "dictionary-based" methods like Lempel-Ziv-Welch (LZW) compression will squeeze it[7]. Finally, the bigger the monochromatic patches (e.g., big expanses of white background) in a graphics file, the smaller a graphics compression technique (like those used in JPEG and for GIF files) will make it.

These facts means that if you send a text file consisting of a high proportion of upper-case characters with PacTOR, you won't get much benefit from Huffman, which relies (in most PacTOR implementations) on a fixed text-character distribution in which certain lower-case characters (like "e") occur with relatively high frequency. Likewise, if you compress a file off-line (e.g., zip it), you produce a compressed (8-bit, or binary) file that looks a lot like a pseudo-random string of bytes. If you then apply a built-in compressor like one from the PKLIB suite used in the P38 CLOVER software from HAL, you will find that the "compressed" file is actually a bit larger than the zip-file. (Of course, this is all right if the zip-ing did a good job.)

On the other hand, if you try to send an uncompressed executable (".exe") image as a binary file with CLOVER II and the HAL software, you'll find that the already pseudo-random structure of most executable (binary) files is likewise expanded rather than compressed by PKLIB. To get efficient transfer by CLOVER II, you should compress executables off-line before submitting them to the HAL, P38 terminal software.

Graphics files (not yet the main focus of our throughput experiments) are another story. If they're GIF or JPEG files, they're generally already compressed, so CLOVER and most other compressors won't make them any smaller*. PICT and BMP files, on the other hand, are not compressed, and often have big monochromatic chunks, so that the

---

[6]The TACO2 protocol suite developed by the DOD for transmission of battlefield-situation graphics over HF has data compression as an integral part.

[7]For this reason, it is not a good idea to compare throughput for (cooked-up) files that consist of repeated sections (for example, those made by repeatedly pasting a section to the end of the file). LZW will generally compress such a file by much more than 50%, whereas Huffman will only compress it by as much as it compresses the first section. The resulting comparison is therefore probably unfair to the Huffman, since in many cases one would send just the first section of such a file with the advice that it is to be repeated N times at the receiver for whatever reason.

[8]The popular shareware EXPRESS terminal program that also runs the P38 and other HAL CLOVER hardware offers built-in compression of files and tailored compression and transmission of graphics images. Since EXPRESS doesn't (yet) fall under our definition of a "common" implementation ("comes with the modem"), we don't cover it here.

PKLIB compressor(s) in the HAL software usually make them a lot smaller, with correspondingly higher throughput.

So far in our NVTS experiments with CLOVER II using both compressed and uncompressed files we have found that compression plays a crucial role in the relatively high average throughput (above 40 characters/s) we report in Table 1. (Recall that this average applies to compressed English text files, and that OHS transfers are not included in our CLOVER data.) The average CLOVER II throughput over NVIS paths for *uncompressed* files is only around 25 char./s, which is about the same as the GTOR throughput of text files[9].

As we pointed out in Section 3, we have not generally sent off-line-compressed files for throughput comparison, so as not to penalize unfairly common implementations of protocols (like AMTOR and standard AX.25 packet) that can't easily handle binary files. The field of throughput comparison using compression techniques that aren't part of "common implementations" is a wide open and important one.

8. Concluding Remarks

One of the conclusions we've reached in our throughput assessments is that hams and others need to separate long from short distance paths when they compare HF throughput performance of ARQ modes, especially the amateur GTOR and packet modes. Some moderation of opinion on HF packet performance may also be called for.

For HF data transfer, data compression plays a crucial role in increasing throughput, and it should always be used when it significantly lowers file or message size and the receiving station is equipped to handle decompression.

We hope that our throughput data will further clarify discussions of the HF digital modes. Our results should put throughput measurements of PacTOR II and other HF data-transmission systems in perspective. We have plans to report someday on the performance of some of those newer modes, and encourage those already in a position to do so to publish their measurements.

Acknowledgments.

I'm grateful to Mike Bernock (KB1PZ), Dennis Gabler (KB5HVN), Doug Hall (KF4KL), Richard Harrison (NT2Z) and Bob Levreault (W1IMM) for handling numerous requests to put their stations on the air in various modes on various frequencies, and for regular mailbox cleanings.

Mike Bernock and Bob Levreault made a number of useful comments on the text of the paper.

References

1. Young, T., et al., "A Preview of HF Packet Radio Modem Protocol Performance," 13th ARRL Digital Communications Conference, Bloomington, Minn. 1994.

[9] As already mentioned, GTOR applies Huffman and run-length compression to all transferred files.

2. Reynolds, P., “HF channel Simulator Tests of Clover,” *QEX*, Dec. 1994.

3. Riley, T., et al., “A Comparison of HF Digital Protocols,” *QST*, July 1996.

4. “HF Throughput Truths,” in Packet Perspectives, *QST*, Feb. 1996.

5. Wickwire, K., et al., “On-air Measurements of HF TOR and Packet Throughput, Part I: Near-Vertical-Incidence-Skywave Paths?” *Digital Journal*, March 1996.

6. Wickwire, K., “On-air Measurements of HF TOR and Packet Throughput, Part II: One-hop Skywave Paths,” *QEX*, June 1996.

7. Nelson, M. and J-L Gailly, *The Data Compression Book*, 2nd Ed., M&T Books 1996.

Table 1. Reliable Protocols for HF Communication

Protocol	Modulation	Bandwidth (Hz)	FECΔ	ARQ	Chan. Rate (Symbols/s)	Character Format	Compression	Avg. Tput (char./s)
NOS TCP/IP+	BFSK	≈ 500	NO	CRC§	≈ 200#	8-bit ASCII	No	2-6◊
AX.25 Packet	BFSK	≈ 500	NO	CRC*	≈ 200#	7-bit ASCII	No	110
AMTOR	BFSK	≈ 500	NO	parity check§	100#	Baudot	No	50◊
PacTOR	BFSK	≈ 500	NO	CRC**	100/200#	8-bit ASCII	Huffman	19◊
GTOR	BFSK	≈ 500	Golay	CRC**	100/200/300#	8-bit ASCII	Huffman/run-len	280
CLOVER II	BPSK-8PSK†	500	Reed-Solomon	CRC***	31.25#	8-bit ASCII	PKLIB (LZW?)	40-500
PacTOR II	BPSK-8PSK	450	Convolutional	CRC**	< 300#	8-bit ASCII	Huff/rn-len/Markov	40-50
CLOVER-2000	BPSK-16PSK¶	2000	Reed-Solomon	CRC***	31.25#	8-bit ASCII	PKLIB (LZW?)	160-200
ALE EOW¤	8FSK	≈ 1800	Golay	CRC**§	125	7-bit ASCII	No	70
FED-STD-10521	BPSK-8PSK‡	≈ 2700	Convolutional	CRC***	2400	8-bit ASCII	TBD	150
SHAPE TC	BPSK-8PSK‡	≈ 2700	Convolutional	CRC***	2400	8-bit ASCII	TBD	150

\# Amateur channel rates are limited by an (outdated) FCC restriction to 300-baud operation at HF. (Shown rates are those settable by an adaptive protocol or those we set for on-air measurements.)
\+ TCP/IP can go much faster with much higher throughput when a more robust modem is used.
A Protocols with interleaving: GTOR, PacTOR II, ALE, FED-STD-1052, SHAPE Technical Centre.
§ Stop-and-Wait ARQ.
0 On-air measurements using text files over actual NVIS or OHS paths.
* Stop-and-Wait or Go-Back-N ARQ.
** Stop-and-Wait with "memory ARQ."
**@top-and-Wait with memory ARQ, up to six retries.
***Selective Repeat ARQ.
† For the P38. Other models can go up to 16PSK and 16QAM (quadrature amplitude modulation). (The "PSK" waveforms are actually four-tone hybrids that switch phases during zero-amplitude intervals.)
¶ Also uses various QAM modulations.
¤ Engineering order wire (not implemented in every ALE system).
‡ Usually employs MIL-STD-188-110A serial-tone modem and sometimes ALE to choose channel.

Notes

1. PacTOR, GTOR, CLOVER, PacTOR II, FED-STD-1052 and SHAPE TC are automatically adaptive.
2. CLOVER II average throughput is probably higher with models that can use 16 signalling elements.
3. Assessed linking probabilities: high (CLOVER II, ALE); moderate (AMTOR, PacTOR, GTOR); low (NOS TCP/IP at night, AX.25 packet at night).