Skimming the Layers

A Survey of Software for Logging and Analyzing the Performance of Modern HF Data Communications

Ken Wickwire KBlJY
232 North Rd #17 Bedford MA 01730,
HF Packet/PacTOR/GTOR: KB 1 JY, KB 1 JY -2, etc .,
SELCAL: OCRS, APRS: KB 1 JY,
HF ALE: MB 1, MB4, etc.,
VHF Packet: KBlJY@WAlPHY,
Internet E-mail: kwick@mitre.org, etc. [1]

1. Introduction

Many parents have recently announced the rebirth of HF radio through the midwifery of digital signal processing. Newer and older hams have discovered or rediscovered the ionosphere as the place where PacTOR, GTOR, CLOVER and APRS hang out. Other amateurs connected with commercisl or government HF are excited about the increasing use of automatic link establishment (ALE), and of data modems with serial- and multi-tone waveforms, forward and reverse (ARQ) error correction, and equalizers. Government HF standards committees are well into the development of sophisticated software for adaptive communications at the Data Link and Network layers of HF data-transmission systems.

With this rebirth has come not just pride, but also bafflement and frustration: what modes should I learn and use? Should I spend my money on GTOR, or try my luck with the more expensive CLOVER? How much faster is PacTor II than AX.25? Does it make any sense to use TCP/IP over HF? When amateurs get ALE, how will I decide if it's worth the cost, and cracking my head over the differences between sounding and scanning and calling and linking?

The answer to each of these questions has to do mainly with ***performance:*** how long will it take to get my file to WlXYZ using PacTOR? Will I clog the 20-meter band with repeated message frames less if 1 use CLOVER instead of PacTor-II? If I buy an ALE modem, will I really link up with DL4ABC faster than I did when I relied on the *How's* DX predictions in QSl? Would a serial-tone modem raise my BBS traffic throughput high enough to justify its hefty price?

It would be nice if we could consult a handbook or call an ELMER or dial up a BBS and get a quick, clear and accurate answer to such questions; perhaps we'll be able to some day. To get an accurate answer at the moment (if not necessarily a quick or clear one) takes a combination of theoretical prediction of a system's performance and on-air data to support and qualify it. I'm going to cover some ways to get that data.

Two phenomena have led to the development of the HF digital hardware and software I'll cover: the random, the time-varying ability of the HF channel to support communications, and the arrival of digital signal-processing techniques that can deal with that variation. However, because of the extraordinary difficulty of characterizing the

[1] The look of things to come? Sooner or later we may have an Internet-type "Universal Resource Locator" (URL) system for amateur digital communications, and an Internet "Web-server for Hams."

non-stationary randomness of the HF channel, which is affected by fading, multipath, noise and interference, theoretical predictions or on-air experience alone can rarely give convincing answers to questions about performance. While most hams are willing to trust the experts where the derivation of predictions is concerned, both the experts and the rest of us expect in the case of HF that theoretical claims will be backed up by *measurements* of on-air performance.

The purpose of this paper is to describe what it means to "measure HF digital performance" on the air, and to give an overview of what's needed (and available) for making such measurements. It will turn out that in most cases the hardware needed to assess over-the-air performance comes with the system to be assessed: if you have the system (usually a computer, a radio modem and an HF transceiver with an antenna), you have all the hardware you need. A surprising amount of freeware or shareware is also available. I believe that hams using the HF digital modes will increasingly see that they need such software to make sense of the river of bits gushing across their screens; I think the days will soon be over when a few sessions with a microphone or key can convince us that a new digital system is worth buying, learning and using.

For some hams the prospect of setting up new software still brings a shudder. I'll try to show them here that there's welcome news about performance-measuring software: although not everybody can or wants to write it, a great deal of very sophisticated- but increasingly user-friendly-shareware for radio/modem control and performance assessment already exists, with more on the way. In addition, as the number of hams getting into digital HF communications increases, so also does the number who are able and willing to write software that compiler-shy hams can use[2].

This paper surveys of some of that software. I hope the survey will lead more hams to think "digitally" about HF, and encourage developers to write more effective packages for assessing the new modes that are arriving faster than most of us can keep up with them. To cover the software, I've had to provide brief descriptions of the various HF digital modes. The paper may therefore also be useful to those looking for an overview of the latest hardware and protocols available for bit-moving on the shortwave bands.

The paper has five parts. The next (second) part discusses the distinction between link and network assessments and the kinds of measurements needed to do them. The details of how the measurements are made in each case may be further broken down according to the waveform, error-control schemes and communications protocols used (e.g., AX.25, GTOR, TCP/IP, Federal-Standard-1052, etc.), and the specific hardware (TNC, serial-tone modem, etc.) that implements the waveform and protocols.

The third and fourth parts discuss performance assessment for particular cases of widely available digital communications protocols (as implemented in single- and dual-port TNC& CLOVER-cards, ALE modems, etc.). Part 3 covers links, and Part 4 networks. These parts contain numerous tables and displays of output from assessment programs. In cases where I don't know of any public software for assessing a particular mode's performance, I'll plant some ideas that developers and manufacturers may want to follow up on to fill the gap.

The fifth part is a look into the future: it treats sophisticated data c.ollection schemes developed for the assessme nt 0f systems that encrypt data and use separate modems for

LThe desire to see one's name appear on hundreds of computer screens is fortunately also hard for hams to resist, linking and data transmission. This part also introduces statistical ideas and techniques that may be found useful in understanding and analyzing digital communications data.

I'm sure I've failed to cover somebody's favorite monitoring system. Readers who know about hardware or software I haven't covered should send me info about it at one of the addresses listed above.

Notes. Although the emphasis of this paper is on the HF band, where transmissions experience the most rapidly changing L and thus, most difficult -channel conditions of any radio band, most of the ideas and software discussed should provoke thoughts on the assessment of performance in other bands.

The paper will not cover contest logging software and "traffic monitoring" packages (like Pawel Jalocha's PKTMONl, the Personal Code Explorer^{TM} , the MFJ Multi-Reader 1 and Kantronic's GMON and its off-line variants) written mainly to allow third parties to read (sometimes without a modem) the communications of connected stations. Although some of these packages are quite sophisticated, their main purpose is either operator-assessment or listener-amusement, rather than system-performance measurement as such. They are therefore outside the scope of this survey, Also not covered is logging of WORLI and other BBS usage, since this monitoring concerns mainly "local" operation and is not generally applicable to non-BBS communications.

2. Link versus Network Performance

To keep the discussion simple, I'll define a *link* as a pair of communicating stations with no intervening relays I define a *network* as a set of three or more stat.ions connected by more than one link.

When we assess **link** performance, we usually have control of both ends of the link and can gather data from at least one of them. This approach, involving only two stations, I'll call *auto-assessment.* It offers the advantages of direct and immediate recording of parameter adjustments and their effects, although recording of requests for retries and numbers of retries is usually difficult. A variant is to use a third station to monitor the performance of two others. I'll call this *third-party assessment.* It offers the contrasting advantages that retries are often easier to record, and that time-tagging and other labeling of monitored packets can often be provided by the third-party's software (for example, in its TNC). In both approaches, we are usually interested only in how the stations on the link cope with channel variations, rather than the details of what other stations are doing and how they affect the link.

In the case of the older, non-adaptive protocols like AX.25, AMTOR and RTTY, we're generally interested in measuring and recording throughput, character-error rates and numbers of repeats. For newer, adaptive protocols like PacTOR, GTOR, PacTOR II and CLOVER, which prescribe *reaLtime* adjustment of protocol parameters in response to changing communications conditions, we might want to record data rates, interleaver depths, frame-lengths, frames per frame-window, numbers of erroneous frames per frame-window, bit- (or character-) error rates, power levels and so on.

In some cases, the values of these parameters can be recorded or inferred directly from the ASCII screen or file output that appears at one or the other of the link stations. In other words, these values are part of the recorded message traffic. In other cases, the software that implements the communications protocol being assessed determines these values, and either uses them to change frame sizes, etc., or sends them to a modem or another peripheral device to adjust its performance. In these cases, recording and analyzing the parameter values is (or should be) an option provided by the protocol software, as is the case with several parameters adjusted by the CLOVER system.

In still other cases, the protocol software does not record its parameter changes, and they have to be recorded by other hardware and software that monitors the control and response lines between the system controller (often a PC) and the controlled equipm$_L$ent. An example of such a parameter is the data rate of MIL-STD- 18% 11 OA serial-tone modem, which is changed by sending an escape sequence and a speed-change command to the modem.

In some systems, parameter values can only be determined after special hardware modifications of the modem. Examples of-this are recording data-carrier detect (DCD) indications from older TNCs, and the average Huffman compression ratio in a PacTOR modem. Data rate changes from 100 to 200 bps and back in a PacTOR modem, and between 100,200 and 300 bps in a Kantronics dual-port KAM with GTOR can be recorded by connected stations themselves using so-called *host-mode* commands (see below), but this may not be possible when monitoring a link between connected stations.

Although manufacturers often provide a means for visual monitoring of data-rate changes (via panel LEDs, etc.), few (with the exception of CLOVER) allow easy recording of these changes. I suggest that the next generation of adaptive modems for HF radio provide this option as a matter of course. Depending on the parameter in question, and whether it is changed by software in the modem or in the PC, parameter changes should be recordable via a dedicated monitoring port on the modem, or as part of the accessible control characters that flow between the controller and the modem.

In assessing **network** performance we are usually interested in numbers like the average throughput of the whole network, the distribution of message delivery times, the sizes of message queues at individual stations, the numbers of relays required to deliver messages and the time it takes a sender to get an acknowledgment of a sent message. The parameters we can adjust to improve network performance, often dynamically in adaptation to changing channel and network conditions, are too numerous to cover completely here. They range from the link-oriented parameters mentioned above, to purely network-oriented ones like message-buffer sizes (a flow-control parameter), routing protocol adjusters and parameters associated with what kind of performance data stations should exchange with each other and how often.

A good example of the latter are the channel quality numbers measured and stored by an ALE modem: if stations exchange channel quality too rarely, the network’s ALE modems may try to link on frequencies that no longer support communications; if they send them too often, the network gets bogged down with the traffic generated by channel-quality exchanges $ to the detriment of traffic throughput.

The best way to assess network performance is to develop software and hardware that monitors the performance of each station in the network, and make sure that the whole network is under the measurer’s control. (An example of such a system is discussed in Section 5.) For amateur networks this is not easy: our networks are big and not under the control of a single group of stations, and our TNCs and multi-mode controllers (excluding CLOVER) were designed for communications rather than data collection. All we can do at the moment is monitor traffic on a local or regional basis and hope that more or less coordinated attempts to tune network parameters (backoff times, packet lengths, etc.) improve performance significantly. I’ll cover available software for doing such monitoring below; more is being worked on.

Let's start with approaches to measuring HF link performance and the kind of software needed to do it. Keep in mind that the point of measuring performance is to give operators a valid means for deciding whether some parameter change (data rate, number of repeats before a data-rate change, etc.) improves performance. (Deciding which parameters to change is not always easy, and figuring out why a change leads to better or worse performance is often even less clear, yet both need to be based on performance data.) The presentation will be organized roughly according to the sophistication of each mode's modulation and error control schemes. For a discussion of some performance results on amateur modes following this approach, see the article by Young et al. in the proceedings of the *13th ARRL Digital Communications Conference.*

3. Link Assessment

I'll start at the beginning, where we find modulation schemes with no error control. The two I'll discuss have been used for decades.

3.1. Morse and RTTY Links

In grouping these, I'll assume that the Morse is sent by hand or automatically, but is decoded automatically. (Human decoding of Morse code involves signal processing and error control that are too complicated to be covered by this treatment.) The Morse "signaling set" is ternary (dot, dash and space, with the carrier turned on and off), while the RTTY set is binary (mark and space with FSK or AFSK modulation). In both cases, the signaling alphabets (transmitted character sets) comprise the 26 letters and nine digits plus various punctuation marks and prosigns. The commonly adjustable parameters are the data rate (and perhaps weighting) with Morse, and the data rate (45-300 baud) and shift (1704350 Hz) with RTTY. These can be changed automatically during a connection with appropriate commands sent, for example, by a terminal scripting language like Crosstalk@.

Although I know of no shareware for auto-assessment[3] of Morse and RTTY communications, writing it would be straightforward. A general approach is to send series of messages of fixed length whose content is known at the receiver, and have receiver software compare incoming characters with the expected ones. Since there's no need to assess performance in real time, this requires only terminal software that can capture screen input and store it in a file. (Most modem terminal programs can do this, but a dumb terminal can't.) Care has to be taken (especially with computer-decoded Morse) to avoid unfair penalization of the decoder resulting from a lost character, since these modes offer no error detection or character counting. That is, the shift of character-position caused by the lost character should not normally lead to the counting of multiple errors. The appendix shows the results of a program that compares a received file with a correct file and assesses their differences in various ways.

It's usually a good idea to add time tags to the logged data. This is commonly done at the beginning of each line of received data if the received stream contains line-delimiting control characters. If the file logging is done via a modern programming language like C, Pascal or FORTRAN, this can be effected by calls to the logging computer's clock;

---

[5] Third-party assessment would generally use similar software.

alternatively, some terminal programs' script languages can add time tags to screen-capture data.

Since the options for improving digital communications via Morse and RTTY are relatively limited, let's leave the assessment of these modes and proceed to

3.2. AX.25 ("Packet") Links

Many parameters can be adjusted to affect the performance of an AX.25 packet link over HF (baud rate, packet length, persistence, number of unacknowledged frames outstanding, number of, and time between, retries, slot time, etc.). However, the mode uses only a 16-bit CRC for error detection coupled with an automatic repeat request (ARQ) protocol for error control. This makes it pretty unsuitable for HF, whose channels often require a combination of forward and backward error correction-i.e., an enror-correction code *and* an ARQ technique- to achieve throughput of more than a few characters per second. As a compensation for the poor performance, the AX.25 protocol supplies a rich set of commands for monitoring and labeling connected and third-party traffic.

Since received packets are free of errors (unless the PASSALL flag is set, which allows display of packets that fail the CRC), auto-assessment of AX.25 packet performance over a link often amounts to estimating average throughput as a function of the parameters that experimenters think are worth adjusting. To assure that no characters from non-linked stations on the channel corrupt the received stream, the CONLIST switch can be turned on to allow only connections with (and data from) stations listed with the BUDCALLS command[4]. (Remember, however, that "suppressed" stations can still cause collisions.)

The measuring station can turn on the MONITOR, MCON and MSTAMP (or CSTAMP) switches to label packets with date/time stamps. MCOM and MRESP can be turned on to monitor and record AX.25 control packets (45, <D>, &A>, etc.) and AX.25 response packets (<FRMR>, <REJr>, etc.) plus sequence labels for sent or received information packets (<Isr> on a KAM). Aside from software carrier-detect logging, which is not provided by standard TNCs, these commands probably provide all one needs for packet performance logging e

Third-party assessment is likewise aided considerably by the software contained in every TNC (for third-party assessment of network performance, see the Section 4) The MON switch allows various kinds of monitoring of channel traffic? The BUDLIST or SUPLIST switches can be used to restrict monitoring on shared channels, and MSTAMP, MCOM and MRESP can be turned on to time-stamp packets or record control packets. To list monitored stations, the MHEARD command with its various options can be used.

The PASSALL command allows display and capture of frames whose starting and ending flags are recognized, but which fail the AX.25 CRC. Here are a pair of captures on 7.09851 MHz LSB on 20 May 1995, with PASSALL OFF and ON:

[4]These are Kantronics commands. In AEA units, similar commands are CFROM, etc.

"11~ some experiments, interference from uncontrolled stations is undesirable, and is avoided by choosing an unused frequency; in others, the effects of interference from other stations are part of the data being monitored, and recording third-party traffic 011 a shared frequency is acceptable.

PASSALL OFF:

```
WQSPUKlRG/H:Pcmer  Module  VHF  ?
R:950518/1824   5176@WSPL.v'r.uS1u?~
R:950517/2039  l@ON7RC.#BR
WAEPISKlRfQG/H:rig.
First1  hadirregulartransmitions onWE', and finaly foundthe final Hybri
WGSPDKlW/H:rig.
First1 had irregular transmitions onVHF, andfinaly foundthe final yrbri
WQSPDKlW/H:d w mdule to be the reason.
Can scmone tell IE the characteristic differen
WQSPDKlw/H:d EXYWEX mdule to be the reason.
```

Here are the same two stations a few moments later with **PASSALL ON:**

```
Hi and many thanks for read this.
##3    ti!&fOanm
WSP>KlRQG/H:ay 16th to 31th at 24:00 UX the special*station
EGlRD will be on the air. This
WQSPWKlRs/H:ay 16th to 31th at 24:00 UTC the special station

Dried herbs: Keep in cool, ark place. They generbloy bffin to lo
WQSP~K~/H:&etwose   not5d.
Dried herbs: Keep in cool, ark place. They gen~ti@&G&&&c&2
U-0Ab~~~@~B&Ult~~se   potency
within 6 llylnths. Crush in fingers to check for arcxna. May be refrige
WSPUKlRQZ/H:se potency
 within 6 nw>nths. Crush in fingers to check for aroma. May be"Qefrige
WA2SPDKlRQG/H:rated if you have rocm.
```

Notice in both cases the repeated frames, and in the second case, the frames (with one or more nonsense characters -here interpreted by a Macintosh) that failed the CRC.

Recent graphical user interface (GUI) terminal programs (and several monitoring and other packages now in the works) use the KISS interface to talk to TNCs, which usually allows faster access to TNC functions and data (compare this with use of the host mode interface, which has a similar purpose). *Savant* for the Mac by Jim Van Peursem (KEOPH) is an example of such a program. Savant's user screen displays, in real time, the number of MAXFRAME packets that have not yet been acknowledged, the number outstanding, the number of retries and the round-trip time. Unfortunately, the current version apparently provides no way to record these statistics.

33. AMTOR Links

AMTOR (and its parent SITOR) is the first widely used system with forward error correction (FEC or "Mode B ," using simple two-fold character-repetition), error detection (based on a check that each seven-bit character has four ones and three zeros) and ARQ (the chirping "Mode A"). The combination of forward and backward error correction, typical of more modem systems, is not provided by the AMTOR protocol.

In the FEC mode, which sounds like RTTY, detection of just one erroneous character in a repeated pair causes the "correct" character to be printed; if both characters are declared erroneous, or both are "correct," but don't match, a "missing character" symbol is printed. (In some TNCs this symbol can be specified, which may make analysis of character errors by a parsing program easier.) This allows for simple (usually off-line) comparison of received characters with sent ones for auto assessment. Throughput and estimated character error rate are the usual measures of performance, and programmed comparison at the receiver of stored copies of sent messages the standard means of assessment.

Third-party assessment of stations communicating in either FEC or ARQ mode is possible. (Some multimode TNCs monitor one mode as a default. The Kantronics KAM "LAMTOR" command allows automatic reception of either mode.) Monitoring at a well-placed location of FEC transmissions (whereby the monitor decodes the repeated characters) or ARQ exchanges might allow a useful independent assessment of which is more effective, since the transmitting and receiving stations have no easy way of automatically recording the number of repeated characters in the ARQ mode. On the other hand, because it is not in dialog with the sender, the third party monitor may entirely miss characters repeated in the ARQ mode.

Auto- and third-party assessment can be aided in the AMTOR, PacTOR and GTOR modes by turning on the TRACE command. This allows display and recording (in HEX) of full header information in addition to information content; in particular, ACKs and NACKs can be monitored with this command set to ON.

Since few parameters can be adjusted to improve AMTOR communications (among them, the mode (FEC or ARQ) and in some TNCs, the delay between receipt of a character triplet in the ARQ mode and its ACK/NACK), the system's further development for HF digital communications is limited. Let's turn therefore to the first HF system to offer *automatic* adaptation to shortwave channel conditions.

3.4. PacTOR Links

PacTor has, like AMTOR, both FEC and ARQ modes, but differs from it in providing automatic adjustment of data rate (100 or 200 baud) and Huffman data compression. PacTOR also allows a variation of conventional ARQ called "memory ARQ," in which settable numbers of repeated but erroneous frames are saved in an attempt to reconstruct a single correct version of the frame. The version of PacTor used by most hams, and treated here, uses binary FSK, like packet. German hams have recently begun marketing PacTOR II, which uses binary and higher-order phase-shift keying (PSK), and differs in some other respects from ordinary PacTOR.

Several parameters can be adjusted to regulate how PacTOR adapts its data rate and ARQ scheme to channel conditions (the data rate can also be set manually). In a KAM (other implementations are similar) these parameters are

- the number of consecutive erroneous packets that cause the data rate to be automatically lowered to 100 baud,
- the number of consecutive error-free packets that cause the data rate to be automatically raised to 200 baud,
- the allowable number of *unsuccess-fd* attempts to increase the baud rate before it is raised automatically only by the previous criterion,
- the baud rate, or the choice of automatic baud-rate selection,
- the number of repetitions of each frame in the FEC mode,
- the number of link attempts or consecutive erroneous frames before time-out and
- the number of erroneous frames stored and used to construct an error-free frame with memory ARQ.

Although manufacturers advise that data rates be set automatically, the fact that many parameters (including data rate) can be set by the user gives wide scope for experiments with adaptive link control. For those who don’t want to tackle serial I/O programming on a PC or Macintosh (not generally for the faint-hearted), powerful scripting languages like Crosstalk@ can be used to implement adaptive control of PacTOR communications.

Auto-assessment of PacTOR performance can be achieved by calculating throughput in the usual way (with calls to the processor clock), by comparing received with stored text in the FEC mode, and by displaying and recording (through screen-capture) the supervisory information exchanged by PacTOR stations. (Some-and perhaps all - current implementations of PacTOR fail to include this supervisory information, and the command to display it is a mere place holder.)

Third-party assessment of both FEC and ARQ communications is possible, and may be aided in some experiments by displaying (when possible) exchanged supervisory information at the monitoring station.

Another monitoring command that’s useful for PacTOR assessment is TRACE, which can also be used in other HF modes. The TRACE command allows display and capture (in HEX) of all monitored frames, with frame data-contents also displayed in ASCII.

Here’s a capture of a PacTOR exchange on 14.07666 MHz LSB, on 20 May 1995, monitored with PTLISTEN, first with TRACE OFF, then with it ON, a few moments later:

TRACE OFF:

```
Software: [33mXP that LL Joachim and get a copy of XPCm... much
much better program than the poor IL... XPCXM and QE
BTUJoachim..
is PK-232 for the KM...
702 S. Ashbrook
On this side of the Atlantic 9 out every using this
program... You should get a copy from look at BWTUPS it needs
registration nne c-tip? Huw does the si FSVRU de YVlAQE..
Mybeamis stuckpointi.ngtotheUSAJoachim, soyoursignal
not strong sounds clean... As S-3
FSVAUdeYVlAI
I hier this evening on 40 m decribing the signal aOk willy, the
NOTHING wrong our and sharp and with other ststions reseption 40
hi Have a nice
Joachim.. Andx
YvlAQE
```

TRACE ON:

```
[AA20202020202020202020202020202020202020BA0D0128E5]
[AA20202020202020202020202020202020202020BA0D0128E5]
[AA20202020202020202020202020202020202020BA0D0128E5]
[5546355641550F0F0F46355641551E]<C:F5VAU>

[AA31797631617165 0C0105A4]~1aqe
[CBCODOAlB5B33316DO32FE8]
[5520202020BlBlBlBlBlBlDC2020BlBlBlBlBlBlDCOO!j384]  +i-c)-tt(  m<
[5520202020BlBlB1BlBlBlDC2020BlBlBlBlBlBlDCOO5384]
[5520202020B1B1B1BlBlBlDC2020BlBlBlBlBlBlDCOO5384]
[AAODOAlB5B33316D2020202020DFDFDFBlBlDB20200lDE57]
[3lm flflfkn
[5520DFDFDFB1B1DB1B5B33356D20205468616E6B7302EB68]  fMfb#[35m  Thanks
[5520DFDFDFB1B1DB1B5B33356D20205468616E6B7302EB68]
[5520DFDFDFB1B1DB1B5B33356D20205468616E6B7302EB68]
```

```
[AAOD260919597AlA5COE2867DA8A5OF8F87C7C3O7DD2C]  Joachim,  for  the  very  n
[556963650D0A1B5B33316D20202020202020B1B1B1004210]ice
C3h
[556963650D~B5B33316D20202020202020B1B1B1004210]
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~-n[-36m   PAC
[552F2806F5lA91829F443757BAl3AB66D887~3E306F8OC]~R   qso   frcm   mrraine
[~B33316D20202020202020B1B1DBDF202020DF03C3E4][3~          4%   fl
[AA42E3520B12B8588D48924E18F6614BB178783C3C05F9DE]an do it again soon.
[AA42E3520B12B8588D48924E18F6614BB178783C3C05F9DE]
[5533316D20202020202020B1B1DB2020B1B1B1B102EE9D]31m        ±±¤  ±±±±
[AAB1B1DB1B5B33326D202054616B6520636172652C03D1A4]tt#[32m  Take  care,
[5509E50D4116C4C9028212B8589D300F8F87C7C3E3040DFF] and please call again.
[55DFDFDFDFDF0D0A1B5B313B33373B34306DC9CDCD02E487]fxKKlf&1;37;40m
[55DFDFDFDFDF0D0A1B5B313B33373B34306DC9CDCD02E487]
[55CDCDCDCDCDCDCDCDCDCDCDCDCDCDCDCDCDCDCDCD002256]
[ [ D C X D C D E ! B O D O A B A 2 O 2 O O l A 9 D O ]
[  [  m  O  D  O  A  B  A  2  0  2  O  O  l  A  9  D  O  ]
[555514433114D32B7142l6OB93A68D61F5E8ElFlFOO6~6D]***   Maracaibo,   Venez
[AADF9008F5BDA50F902869DFCA228AA1188A79783C07FDA8]uela  -  South  Amrica  *Jr*
[555B306DODOA4164696F7320lElE~lElElElElE~OO77DD][~   Adios
[AA4A6F616368696D2E2E2E201E1E1E1E1E1E1E1E01CF18]Joachim...
[551E1E1E1E1E1E1E1E1E1E1E1E1E1E1E1E1E1E1E022476]
[5514EB1814285F1F08F8E8550156F1300B901E1E0F045F2C]YV1AQE terminating with
[AA33A88E4139F864C2BC674980D0428BE268A3DBE0051431]F5VAU.. SK at 22:36 U
[5554432E2E0D201E1E1E1E1E1E1E1E1E1E1E1E1E02A5BA]TC..
[5554432E2E0D201E1E1E1E1E1E1E1E1E1E1E1E1E02A5BA]
[AA1E1E1E1E1E333303303C33330F3C3033331E1E1E832D70]<D:F5VAU33>
[AA1E1E1E1E1E333303303C33330F3C3033331E1E1E832D70]
[AA1E1E1E1E1E333303303C33330F3C3033331E1E1E832D70]
```

Automatic recording of data-rate changes when PacTOR is in its automatic baud-rate-adjusting mode is not possible when the modem is controlled from a conventional terminal program. This would require (on a KAM, for example) monitoring and recording the state of the LED that displays baud-rate switches. (The KAM and other TNCs can, however, be interrogated via their serial ports for data rate, PTT status, sending and receiving status, FEC vs. ARQ mode, etc., when they're in the *host mode.* KGOLD, XPKAM and other host mode applications allow such interrogation.)

3.5. GTOR Links

GTOR also has FEC and ARQ modes, and provides automatic adjustment of data rate from among 100,200 or 300 baud. The (pure) GTOR FEC mode is the same as the AMTOR FEC mode (repetition of characters) and is used for broadcasting. For its ARQ mode, GTOR applies Golay forward error-correction and block interleaving to transmitted data. At the receiver, the data are de-interleaved and checked for errors with a CRC. If errors are found, retransmission of the data is requested. The receiver analyzes the retransmitted data for errors and applies Golay error-correction to them if necessary. If this doesn't correct all errors (it often does), another retransmission is requested. This is an example of what's called *uduptive ARQ.*

GTOR can apply three kinds of data compression to lower the number of bits used to send characters, and can be set to tolerate a small number of errors in frame acknowledgments.

As with PacTOR, several parameters can be adjusted to regulate how GTOR adapts its data rate and ARQ scheme to the channel (the data rate can also be set manually). In a KAM these parameters are

- the number of consecutive erroneous packets that cause the data rate to be automatically lowered to 200 or 100 baud,
- the number of consecutive error-free packets that cause the data rate to be automatically raised to 300 baud (no automatic increases to 200 baud occur),
- the allowable number of *unsuccessful* attempts to increase the baud rate before it is raised automatically only by the previous criterion,
- the baud rate, or the choice of automatic baud-rate selection,
- the allowable number of erroneous bits in an ACK ("fuzzy" ACKs) and
- the number of link attempts or consecutive erroneous frames before time-out.

The fact that many parameters (including data rate) can be set by the user also allows adaptive link control of GTOR communications. Scripting languages are again an easy way to implement adaptive control of GTOR links

Auto-assessment of GTOR performance can be achieved by computing throughput and by interrogating the serial port for data rate, PTT status, sending and receiving status, etc. Host-mode programs can apparently do this already.

Third-party assessment of GTOR ARQ communications is not possible with a standard terminal program because the KAM's processor is not fast enough to distinguish and decode packets in real time[6]. Kantronics has produced a program called GMON that uses the processing power of a 286- or faster PC to assist in monitoring the traffic between connected stations. This traffic is delivered to GMON in the form of "scanned" samples of data from the KAM. On slower PCs recorded scanned samples can be monitored off-line with a program called GOFF. GMON does not apply error correction to received data, and allows in that sense comparison of error-corrected and non-corrected transmissions. A third program, GMONITOR, allows off-line analysis of error-corrected data.

3.6. CLOVER Links

CLOVER is the most advanced adaptive modem sold on the amateur market. The modem modulates and demodulates successive pulses at four pulse frequencies with any one of seven waveforms, including BPSK, QPSK, SPSK, 16PSK and combinations of 8PSK with two amplitudes and 16PSK with four amplitudes. (The latter are called *quadrature amplitude modulation,* or QAM.) The four-pulse waveforms occupy no more than 500 Hz of bandwidth.

CLOVER provides -like AMTOR and PacTOR - an FEC mode and an ARQ mode. In the ARQ mode- the main adaptive CLOVER mode-, the modem automatically chooses the "best" waveform in accordance with measured channel conditions. Since different bit rates are associated with different waveforms, the modem has an adaptive data rate. Its channel rate (data bits plus overhead bits) can be set (normally automatically) between 31.25 and 750 bps (with data rates up to about 500 bps). The CLOVER *symbol rate* is constant, and is 31.25 baud (symbols per second). Using signal-to-noise ratio measurements at their receivers, two CLOVER stations in the ARQ mode can also adjust their output powers for most efficient channel usage.

[6]The v&O KAM software update is said to remedy this shortcoming.

Forward error correction is provided by a Reed-Solomon code-a powerful, non-binary (symbol) block code that can correct burst errors. Codeword lengths of 17,51,85 and 255 bits may be chosen. The code's rate (information bits/total bits) is adjustable to 0,6, 0.75,0.90 and 1 .OO, with corresponding capabilities of correcting codewords containing 20,12,5 and 0% incorrect bits.

Backward error correction in very poor channels is provided by an ARQ approach that uses the error-detecting capability of a check-sum and *selective repeats.* This efficient, but software-intensive approach to ARQ involves requests for repetition of only the erroneous frames in a "frame-window" (set of transmitted frames)?

The modem can collect (and pass on for display and processing) channel-quality and labeling information in the ARQ mode as follows :

- time tag,
- sende&or receiver's callsign 4
- modulation format,
- FEC coding rate,
- throughput (bytes per second),
- SNR (dB),
- frequency offset (Hz),
- phase dispersion,
- used Reed-Solomon FE@ capacity and
- transmitted power.

(In the FEC mode, the receiver's callsign is not recorded, and the coding rate and transmitted power are fixed.) The information listed can also be recorded automatically to a file. The recording format is the comma-delimited line of strings

[time],[call],[modul.],[code rate],[data rate],[SNR],[freq. offsetJ,[ph. disp.],[FEC cap.],[pwr]<CR><LF>

This facility meets most requirements for auto-assessment. Additional off-line analysis of performance can be performed with a spreadsheet or graphing program, or with other parsing and statistical software that processes the time-tagged statistics files.

If only SNR and phase distortion data are required, two CLOVER stations can put themselves in the manual ARQ mode, choose BPSK modulation and send no message data. The stations will then function as a "bi-directional oblique-incidence sounder-system," exchanging only data on signal quality. (See the discussion of ALE systems below for other low-cost HF channel-sounding possibilities .)

Third-party assessment of CLOVER- is possible at a CLOVER-equipped listening station. Both FEC and ARQ transmissions may be monitored-provided that the listener has correctly received the appropriate connection blocks [8] - , and Reed-Solomon error-correction of monitored data blocks can be performed. However, because of the tight coordination of sending and receiving stations required for the selective-repeat ARQ mode, error correction via repeated transmissions is not possible at third party stations.

[7]AMTOR and PacTOR use so-called "Stop-and-wait" ARQ, and Packet uses so-called "Go-back-N" ARQ. Both approaches generally lead to lower throughput than Selective-Repeat ARQ.

*If the connection block (which contains the caller's callsign and announces the waveform) is missed, the listener has to wait for the next connection block, which may take 20-30 seconds.

In many respects, the CLOVER statistics-recording scheme is a model for what future adaptive HF digital communications systems should provide. The data collected by CLOVER offer fascinating opportunities for experimentation and performance improvement. I hope that other vendors and developers will agree, and include corresponding recording capabilities as options in their own products.

3.7. TCP/IP Links

TCP/IP communications over HF are possible using various versions of NOS for PCs or of NET/Mac for the Mac. The software uses a TNC running in the KISS mode. TCP/IP communications are very slow because of the 300-baud HF data-rate restriction, the overhead caused by network- and transport-layer header information and an error-control scheme (ARQ but no FEC) that is not tailored to HF radio operations.

In addition to the standard IP-controlled communications at the network level, AX.25 and NET/ROM connections can also be established from within amateur TCP/IP applications. In each case, TCP and IP packets are actually imbedded in AX.25 or NET/ROM frames for transmission over TCP/IP links or networks. The slowness of TCP/IP over HF is a pity since amateur implementations of TCP/IP offer, with their Internet relatives[9], telnet and FTP sessions, binary file transfers, pinging, SMTP, POP mail and the provision of extensive performance monitoring.

The parameters that can be adjusted to tune TCP/IP performance are too numerous to list here in their entirety. Among them are: various AX.25 parameters (paclen, etc.), NET/ROM timing parameters, maximum number of hops before a packet is discarded, maximum segment (read frame) size, maximum frame-window size and virtual circuit (link and IP level CRCs and ACKs) vs. datagram (IP level CRCs and ACKs only) mode. For HF operation many of these parameters should have much different values than they have for wire or line-of-sight VHF or UHF operations.

Numerous statistics on connection status and communications performance are provided by TCP/IP. Among the most interesting and useful are the initial and subsequent round-trip ping times, which measure the time (usually in milliseconds) it takes a packet of settable length to reach a called station and be returned. Here are the results of pairs of pings (sent at 1200 bps) over a VHF/UHF link via the gateway WAlPHY to W 1 I-MM-2 (40 miles away) and over a VHF link to WA1 PHY itself (three miles away):

```
Ockers> ping wlimm-2
Ockers> 44.56.8.102: echo reply id 0 seq 2052, 11800 ms
Ockers> ping wlimm-2
Ockers> 44.56.8.102: echo reply id 0 seq 2206, 6000 IW
Ockers> ping walphy
Ockers> 44.56.4.1: echo reply id 0 seq 2403, 2800 ms
Ockers> pi ng wa lphy
Ockers> 44.56.4.1: echo reply id 0 seq 2466, 5100 m5
```

The first ping to W lIMM-2 took longer than the second because a suitable route to WlIMM-2 had first to be found; the second ping to WA1 PHY took longer than the first probably because WA1 PHY was busy during the second. Suitably calibrated ping times

[9]1'm told that we can expect within the year an AX.25 implemention that uses the standard TCP driver of a popular PC operating system. This would allow hams to use several TCP performance-monitoring programs to monitor packet-radio communications.

might be useful over HF links as an aid in propagation studies (perhaps of surfacewave vs. skywave modes).

The KA9Q TCP/IP implementations used in ham radio also provide statistics on AX.25, NET/ROM, IP and TCP protocol connections. Here are examples of queries (during a cross-connected VHF/UHF telnet session between KBlJY and WlIMM-2 as to the status of AX.25, NET/ROM, IP and TCP connections:

```
Qckers) ax status
    &AXE IF   Snd-Q   RW-Q   Rmo te    stch
  237902 axQ  0       0   WA   IPHV-6   Cmtwc   ted
Ockw5) nr status
Iriterfac~  SndQ RcwB NunReceived CSurnErrws
#ckws> ip status
IP:  total 26 runt 0 Im err 0 verE; err 0 chksum err 0 badprcrto 0
ICMP: chksum err 0 t-m space 0 imp Q bdcsts 0
type  t-cud  sE37t
Ockerrs> tcp status
cmout 1 conin 0 reset out 0 runt 0 chksum err 0 bdcsts 0
 #    &TCB Rev-Q Snd-Q Lma I socket          Rem te smke t        state
 0  2Qf390    0     0             kbljy.9          #.#.#.Q:Q       Li5tm CS>
 4  2Qf14c    0     Q            kbljy:21         Q.Q.Q.Q:Q        Listen (9
 6  2QfQ88    0     0            kbljg:23          #.Q.Q.o:Q       Listen CS>
 9  2Ocdc8    0     0     kbljy:     1824     wl     imm-2:23      Establ ished
    20f2dQ    0     0            kbljy:?          Q.Q.Q.O:CJ       Listen CS>
 a  2Qf458    0     0            kbljy:79          Q.#.Q.Q:Q       Listen CS>
IQ  2Qf2Qc    0     0            kbljy:25          Q.Q.#.Q:Q       Listen <S>
-   D   I
```

All of these (here either empty or rather uninteresting) statistics might be useful in assessment of HF communications that use TCP/IP-frames encapsulated in AX.25 frames.

NOS and NET/Mac also have a TRACE (and record) capability that can be used for third-party assessment of AX.25 (packet), NET/ROM and TCP/IP communications over HF (and VHF). The TCP/IP TRACE command can be tailored to display output or input headers and the information content of frames in ASCII or HEX or both. Here's an example of some packet traffic on 14.103 MHz monitored using the "trace headers only" version (TRACE ax0 011):

```
Mar 19 18:24:50 ax0 recv:
Ax25: w9yHy-15'N8EiT-7 RR(P) NR=7
Mar 19 18:24:52 ax0 recv:
AX25: W9YJZY-15->N8ETlJ-7 I Mi=7 NS=O pid+Ikxt
Mar 19 18:26:27 ax0 recv:
Ax25: W9YHY-l$-~N8E?T-7 RR(F) NR=O
Mar 19 18:26:34 ax0 recv:
AX25: KD4NDH-13-XDlMT M(P)
Mar 19 18:26:40 ax0 recv:
AX25: KD4NDH-13-XDlMT sABM(P)
Mar 19 18:26:44 ax0 recv:
```

Here's some traffic on 7.098 MHz with both headers and information content displayed (TRACE ax0 111):

```
tkr 20 21:12:36 ax0 recv:
Ax25: NlGMU-l+NODES UI pid=NET/RoM
NET/ROM Routing: WVT
```

```
          NlGMU  BESXJ     Nl@!KJ-1  30
          vJQRM   w        vE2RM      10
        W2Uxc-1  PLE3      W2Uxc-1  10
       mmxf-1  NWVT        WA2SPI.A  10



Mar  20  21:13:42  ax0  recrv:
AX25:  NlGMJ->Klw  I(P)  NR=3  NS=4  pid=Tkxt
0000   .Tx.USA..the  computer.  The  "Y"  cable
Mar 20 21:13:44 ax0 rear:
AX25:  NlGMU->KlRw  I  NR=3  NS=5  pid=!kxt
0000  is  NOT  needed..What  you  do  need  is
```

Note that these trace data contain a NET/ROM routing table and some ASCII text.

Here, for comparison, is a screen capture of some HF packet communications on 21.097 MHz with the TNC commands MON, MCOM, MRESP, MSTAMP and PASSALLPID ON:

```
W4Dl?HXElAIC/H  [20-03-94  19:01:28]:  <<c>>:
W4DPHX7ElAIC/H  [20-03-94  19:01:39]:  <<(3>:
W4DPH>VElAIC/H   [20-03-94   19:02:28]:   =rrO=:
W4DPHXJElAIC/H   [X&03-94   19:02:34]:   =rrO>>:
W4DPH>VElAIC/H  [20-03-94  19:02:45]:  <QxO>>:
W4DPHXElAIC/H [X&03-94 19:03:37]: =rrO=:
N7JOR>K40=-3/H   [20-03-M   19:13:00]:   <<uA>>:
N7JOR>K40LX3-3/H  [20-03-M  19:13:02]:  =IOO=:{FO}~,  NAME  HERE  IS
*EmIEF
```

Since TCP/IP trace data can be written to files, further off-line analysis of trace data is easy. There is a TRACEONLY version of the TCP/IP trace command that allows recording of only TCP/IP packets to or from a particular station. This is the equivalent of the packet BUDLIST ON command (with one station in the BUDCALLS list) for third-party assessment of TCP/IP performance.

The NOS AT Command

Recent versions of the NOS TCP/IP application for the PC have a command called AT (for automatic timing) that allows one to schedule NOS actions like PING, FINGER and STATUS. This useful command allows TCP/IPers to design propagation and communications experiments in which one station automatically interacts with one or more other ones at regular intervals. The same command can be used at sending or receiving nodes to query NOS on the status of attached ports and the statistics of communications at the link, network or transport level (AX.25, II?, NET/ROM or TCP status). If the results of AT commands can be recorded in some way (session recording, screen buffer capture or recompiling NOS to send AT output to a file), one could avoid the use of more complicated scripts or lower-level code in performance assessment.

For example, to ping KSXYZ every five minutes and record the results, one could send the command

AT NOW+0005 “PING KSXYZ +”

followed by an appropriate recording command. (Pinging can, of course, be scheduled with the ping command itself.) In a two-way communications experiment involving scheduled calls, the called station might record the results of the command

```
AT NOW+0020 "TCP STATUS +"
```

which queries NOS every 20 minutes on the status of communications at the TCP layer?

3.8. Federal-Standard-1045 (ALE) Links

Automatic link establishment (ALE) may be the most powerful and interesting development in HF since SSB. The protocols for it were standardized in the late 80s and ALE equipment has been commercially and widely available for about six years. Since ALE requires complicated control of receiver scanning, an ALE modem is usually mated to an HF transceiver as a built-in accessory, but stand-alone models also exist. Prices have been slowly but steadily falling, and there's a good chance that hams will soon be able to afford and use ALE on the HF bands.

The ALE waveform is S-ary phase-continuous FSK. The mode's error-control was designed for very high reliability at the expense of throughput-a normally inescapable tradeoff in the case of HF. Error-control is provided by a combination of Golay forward error-correction coding, interleaving and three-fold bit repetition[11] o At the receiver a majority vote on the repeated bits, de-interleaving and Golay decoding usually assure that there are no errors in ALE words delivered to the communications terminal.

In an ALE network, idle stations are usually scanning a programmed set of between five and ten frequencies, waiting for calls, At regular intervals during the day an agreed-upon subset of stations broadcast short sounding transmissions on each of the frequencies. Any scanning station that hears a sound on a frequency automatically records the address of the sender and the quality of the signal. A special sounding variation called a link quality assessment (LQA) exchange allows a particular pair of stations to measure and exchange on-air quality measurements of the HF channels between them.

These stored channel qualities, if broadcast (or exchanged) by the right stations and recorded often enough (three or four times a day is usually enough), allow network stations to choose the best frequency for communication with any other stations at any time. While some discipline (or restrictions) will have to be imposed on ham sounding if a large number of us use ALE, the technique clearly offers exciting possibilities for great improvement of amateur communications over HF.

When a station wants to call another, it automatically looks through its channel-quality memory for the frequency with the highest quality. It then calls the desired station on that frequency. If the called station hears the call and recognizes its address, it answers. If he hears the answer, the caller completes the "three-way handshake" with an acknowledgment of the called station's answer The two stations are then linked, and may communicate using voice or data (Packet, PacTOR, TCP/IP, etc.). Several more complicated kinds of calls to multiple stations (in nets that expect to be called, or "on-the-fly"), to stations in other networks (ANYCALLS, ALLCALLS), and to stations with common address characters (WILDCARD calls) can also be made.

The software built into an ALE modem (but unfortunately not yet standardized across manufacturers) allows extensive performance monitoring and channel assessment. All

---

[10]AT-scheduled sessions are turned off with an AT K N command, which "kills" session N.

[11] An optional orderwire (text-transmission) mode adds an ARQ technique for further error control.

implementations allow automatic, scheduled sounds, storage of LQA scores and recall of the addresses of called stations with their associated LQA scores. Commands to rank all channels used by a sounding or communicating station allow regular comparison of channel qualities.

Here are some captured two-way channel rankings taken with a 125-watt, ALE-equipped, Harris RF5000 transceiver for channels between Bedford, Mass., near Boston, and NFK in Norfolk, Mass. (40 miles), MT2 in Reston, Va. (350 miles), and MX1 in Ft. Wayne, In. (800 miles). The remote radios also transmitted about 125 watts. The recorded "scores" are combinations of the "measured" SNR (received at Bedford) and Bedford's SNR (received at the distant station and sent as part of an LQA exchange back to Bedford). The largest possible score is 120, and a received SNR of 3 1 in the case of MT2 and MX 1 indicate that MT2's and MXl 's data-although compliant with the ALE standard-did not conform to the format used by the RF5000 for complete display of two-way measurements. This is a common problem resulting from the lack of standardization of measurement disr>lay and score calculation.

```
12/15/94
4:16:00 PM (Gm)
RANKNEK
CHAN: 08 ScylRE: 015   MEmuREDsNR10  REcErvEDsNR30
CHAN: 07 SamE: 007     MEzsumDsNRo7  REcEIvEDsNR17
CHAN: 06 SCORE: 003    MEAsuREDsm15  REcIEzvEDSNR26
CHAN: 05 SCORE: 001    MFAsmEDsNR10  REzEmEDsNR13

12/15/94
4:16:37 PM
RANKMT2
CHAN: 07 SOORE: 095    rmsuREDsNR3oREcEmsNR31
CHAN: 09 Sam: 040      msNR14  REcEIvEDsNR31
CHAN: 08 SCORE: 039    MEAsuREDsNRl4REcEIvEDsMI31

12/27/94
20:22:44 PM
RaNK Mxl
CHAN: 10 SOORE: 085    IaAsmEDsNR21  REcEmsNR31
CHAN: 09 SCORE: 075    klmsuRmsNRl5REcEmsMI31
CHAN: 08 SCORE: 071    ImAsuRmsNRl4REcEIvEDsNR31
CHAN: 11 SOORE: 059    MEAsuREDsNRo7REcEmsNR31
CEWN: 07 Sam 057       MEASUREDsMIO8 mCErvms?m31
```

Auto-assessment of ALE-assisted communications or channel quality can be accomplished using a script language or a lower-level language like C to send commands and queries to the modem. Third-party assessment requires an ALE modem and receiver for detecting and decoding ALE transmissions. The third party can record sounds, or carry out LQA exchanges with particular stations in the scanning mode. Because of the brief time a third-party listener spends on a channel before returning to scan if he does not recognize his own address in a call, third-part monitoring (especially of message traffic) is normally performed only on fixed channels (i.e., in the ALE single-channel mode).

3.9. Federal-Standard-1052 (HF Packet Radio via Serial-tone modem) Links

Federal standard-1052 is still in development, but should be published within the year. The packet protocol part of the standard assumes that a suitable communications frequency has already been found, and prescribes more or less continuous "negotiation" between pairs of stations on that frequency that is similar to what happens with the

CLOVER protocol. The negotiation starts with a linking exchange and is maintained with negotiation packets called “heralds” and “herald-ACKs .” The protocol’s error - control is provided by selective-repeat ARQ, which is the most efficient of the non-hybrid ARQ techniques.

Although 1052 does not prescribe a specific ALE technique, it is often implemented in systems that also use Federal-Standard ALE. No data modem is prescribed by the 1052 packet protocol either, but it is generally implemented with a Military-Standard-l 88 - 11OA serial-tone modem. This modem (currently too expensive for most hams) uses convolutional forward-error-correction coding, a variable-length interleaver, a narrowband (<< 3 kHz) interference excisor and a channel equalizer that uses training frames interspersed with data frames to compensate for waveform distortion caused by HF multipath. The combination of the serial-tone modem with 1052’s ARQ protocol probably provides the most effective data transmission available for the HF channel[12].

Since 1052 is still in development, assessment of its performance is beyond this paper’s scope. On the other hand, the serial-tone modem that usually does the dirty work for the protocol has been around for nearly a decade. The modem has a fairly simple command set (basically TEST, OPER, set INTERLEAVER and set DATA RATE). A higher-level protocol like 1052 uses these modem-commands to adapt itself to channel changes.

The serial-tone modem has (like CLOVER) the ability to measure and send to Data Terminal Equipment (PC) almost continuous measurements of post-processing received signal-to-noise ratio (i.e., after interference excision and equalizing). These measurements, produced at the rate of two or three per second, can be recorded by a terminal program with screen capture, and are a part of auto-assessment of performance. Although they have to be analyzed with some care because they represent SNRs *ufter* signal processing, these data provide an easy and useful means of studying HF fading and interference. Here’s an example of some control-line output from a serial-tone modem. (Message data output can also be monitored.) The control data were gathered during recent daytime reception of on-air message data at 4.5 MHz from a station about 250 miles away. The raw ASCII control data (not shown here) have already been processed off-line by a parsing program that has removed blank lines and labeled status flags (“ready bit on 1)’ etc.).

```
Inpplt file is [02012053.IPM].
Parsed output:

20:58:18 [14;6f[KOPER
21:06:14 [16;9f[K    [19;9f[KOlOO   ready bit on
21:06:24 [16;9f[K  --  [19;9f[KOlOO   busy channel on
21:06:25 [16;9f[K  --  [19;9f[KOlOO
21:06:25 [16;9f[K  --  [19;9f[KOlOO
21:06:26 [16;9f[K  --  [19;9f[KOlOO
21:06:26 [16;9f[K  --  [19;9f[KO100
21:06:26 [16;9f[K  09[19;9f[KOlOO
21:06:27 [16;9f[K  09[19;9f[KOlOO
21:06:27 [16;9f[K  09[19;9f[KOlOO
21:06:28 [16;9f[K  09[19;9f[KOlOO
21:06:28 [16;9f[K  10[19;9f[KOlOO
21:06:28 [16;9f[K  10[19;9f[KOlOO
21:06:29 [16;9f[K  10[19;9f[KOlOO
21:06:29 [16;9f[K  10[19;9f[KOlOO
```

[12]HF Data-link protocols developed recently by the NATO SHAPE Technical Centre in Holland, and by the airlines for automatic position reporting of ocean-flying passanger aircraft, use serial-tone modems to produce similar performance.

```
21:06:29  [16;9f[K  10[19;9f[KOlOl    synch bit on
21:06:30  [16;9f[K   11[19;9f[KOlOl
21:06:30  [16;9f[K   11[19;9f[KOlOl
21:06:31  [16;9f[K   11[19;9f[KOlOl
21:06:31  [16;9f[K   11[19;9f[KOlOl
21:06:31  [16;9f[K   11[19;9f[KOlOl
21:06:32  [16;9f[K   23[19;9f[KOlOl
21:06:32  [16;9f[K   23[19;9f[KOlOl
21:06:33  [16;9f[K   23[19;9f[KOlOl
21:06:33  [16;9f[K   23[19;9f[KOlOl
21:06:33  [16;9f[K   23[19;9f[KOlOl ...  [etc.]
```

(“Busy channel on” marks the first of a sequence of “--” flags from the modem that indicate initial detection of a serial-tone signal. These indications can be used by a system controller to implement a collision-avoidance scheme.) Further off-line processing of the raw data file produces a tab-delimited file of SNRs that can be plotted or analyzed further:

```
T h           j      SWjl (W
21:06:26      1      03
21:06:27      2      03
21:06:27      3      CB
21:06:28      4      (3
21:06:28      5      lo
21:06:28      6      lo
21:06:29      7      m
21:06:29      8      lo
21:06:30      9      11
21:06:30      u)     ll
21:06:31      IL     ll
21:06:31      12     ll
21:06:31      13     IL
21:06:32      14     n
21:06:32      15     23
21:06:33      16     23
21:06:33      17     23
21:06:33      18     23 ...  [etc.]
```

Third-party assessment of serial-tone performance is feasible if the monitored data are not encrypted or otherwise protected by special codes from decoding by listening stations.

3.10. Host-Mode Monitoring of TOR Modes

The host mode in TNCs like the Kantronics KAM and AEA PK-232 facilitates the exchange of commands, responses and data between a PC controller and a TNC in a way that uses simplified (and therefore user-unfriendly) syntax. This allows more rapid (and from a programming standpoint, simpler) control and data exchange. The main application of the host mode so far is in “host” applications that provide easy handling of multiple connected streams, key assignment of frequently exercised commands, help screens, and so on. However, in HF communications using the KAM or PK-232, the host mode offers additional advantages for performance assessment. This requires writing monitoring software that takes advantage of host-mode commands, and doing so should be encouraged.

Host Mode programs often provide the user with data on communications performance (with a connected station or as a third-party listener) that are easier to interpret than flashing LEDs, etc. They do this by sending the TNC special hostmode “query” commands whose responses give nearly-real-time data on PacTOR (or in the case of the

KAM, presumably also GTOR) data rate, Huffman coding, TOR receiving or transmitting state, number of retries and outstanding frames, etc. The host mode application converts the user-unfriendly responses to friendly screen displays of the corresponding status messages.

All frames exchanged between the controlling computer and TNC are delimited by fixed characters that tell the receiving end of the exchange that host-mode commands or data are arriving. In the case of the PK-232, host-mode exchanges are delimited by the Start of Header (SOH, hex 01) and End of Transmission Block (ETB, hex 17) characters. For the KAM, hostmode exchanges are delimited by the Frame End (FEND, hex CO) character.

For the PK-232, host-mode queries from the PC to the TNC have the form

SOH A B ETB,

where A and B stand for a two-letter mnemonic of the corresponding "verbose" commandI . The PK-232's response has the form

SOH hex4F a b (value) ETB,

where a and b form the query mnemonic and (value) gives the data of the response (Y, N or numerical data).

For the KAM, host-mode queries from the PC to the TNC have the form

FEND?FEND,

where FEND is hex CO. The KAM's response frames have the form

FEND?OMSXYFEND ,

where ?O indicates a response to a query, M gives the mode (Packet, FEC, PacTOR, GTOR, etc.), S the "Submode" (connected, disconnected, standby, etc.), X the "Status" (idle, failed CRC, received request for repeat, Huffman compression, data rate, etc.) and Y shows if PTT is active or that a changeover of transmitting and receiving stations is in progress.

These host-mode queries can be invoked by monitoring programs that can inquire after communications status and performance in connected and some monitoring modes. The queries can be sent on schedule or in response to changes in performance indicated by query responses, throughput, numbers of errors, etc. The responses can be logged for further analysis via screen capture or by having the monitoring program itself write them to a file.

4. Network Assessment

I've defined a network as a set of at least three HF stations, which often means that stations can use automatic relaying. Although auto-assessment of performance by

1[3]The PK-232 can also send queries about the current operating mode (Packet, AMTOR, FEC, etc.j and link status (AX.25 vl or v2, number of unacknowledged packets, retries, etc.). A Data Polling command allows one to query the PK-232 on a regular basis as to its status, whether it has changed or not.

stations involved in true HF network communications, rather than "potential" networking (see the discussion of the IPS in Section S), is possible, I don't know of any networks that are currently doing it on a regular basis I[4]. Most of this section will therefore cover third-party assessment.

4.1. AX.25 Networks (Monax25, PACKHACK, PacketTracker, MacAPRS)

Since AX.25 packet radio is one of the oldest data transmission modes used in HF networking, most of the software for third-party assessment of networks pertains to monitoring of AX .25 traffic.

4.1.1. Monax25

Monax25 (for "Monitor AX.25") is a set of C-programs written in the late 80s by Skip Hansen (WBGYMH) and Harold Price (NK6K), which allow real time monitoring and recording of AX.25 packet traffic on one frequency. The programs were written to allow "global" assessment of VHF or HF LAN traffic patterns, where the "local" in LAN means "within receiving distance ."

All the Monax programs use the DOS command-line interface. The real-time monitoring is performed by a program called STATS.EXE which has fast, assembler-language modules that handle serial-port I/O with a TNC operating in the KISS mode. STATS displays packet, NET/ROM and TCP/IP address) data and control fields, including the retry flag, in real time on the standard output (screen), and dumps summary data to a time-tagged log file every five minutes (this interval can be easily modified by changing and recompiling the available source code). STATS can also record data-carrier-detection (DCD) activity (monitoring true DCD activity requires a slight modification of older TNCs). Here's a snippet from the beginning of such a log file, recorded in February, 1995, on 14.105 MHz:

```
T,791943670
F,0,0,0,0,0,0,0,0,0,0,4268
T,791943972
F,4,75,3,58,4,0,0,0,0,0,5172
C,VE1CRS-7,N9BMS,0,240,1,1,1,4,4,6,6,6,75,75,791944053,0,0,0,0,0,3,0,1,0,0,0,2,1,0,0,0,0
T,791944274
F,27,503,16,310,26,1,0,0,0,0,5175
C,W9SL,WG9V-7,0,0,0,0,0,3,3,0,0,0,51,51,791944285,0,0,1,0,0,2,0,0,0,0,3,0,0,0,0,0,0
C,KI5DQ-7,KA9ALO,0,240,1,3,3,14,14,2,6,6,247,247,791944322,1,0,0,0,0,10,0,3,0,0,9,2,1,0,0,0,0
C,BEACON,N9BMS,0,240,1,1,1,1,1,34,34,34,52,52,791944350,0,0,0,0,0,0,0,1,0,1,0,0,1,0,0,0
C,KA9ALO,KI5DQ-7,0,0,0,0,0,1,1,0,0,0,17,17,791944367,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0
C,W4KBS,KC4FS,0,0,0,0,0,6,6,0,0,0,102,102,791944492,0,0,1,0,0,5,0,0,0,0,4,1,0,0,0,0,0
C,VE1CRS-7,N9BMS,0,0,0,0,0,2,2,0,0,0,34,34,791944510,0,0,0,0,0,2,0,0,0,0,1,0,0,0,0,0,0
```

The labels at the beginnings of the log-file lines stand for T(ime), F(requency) and C(ircuit). A circuit is an AX.25 Level-2 connection between a pair of stations. Time records appear about every five minutes. Frequency records follow the time records and contain total packets, bytes, unique bytes, and several other statistics on traffic on the monitored frequency during the previous five minutes. Circuit records contain, for each connection monitored during the previous five minutes, the to- and from-calls, total data packets, time the circuit was monitored, the numbers of packets of various sizes, etc. (For a detailed rundown of the recording format see the documentation that accompanies

1[4]1'd be happy to hear of examples, perhaps fkom operators of PacTOR, CLOVER or GTOR BBS message-forwarding stations.

the program archive, which can be found on various bulletin boards and Internet FTP sites .)

The log file can be analyzed off-line with other programs. The next one that's normally used is called REPORT, which combines and summarizes log information to produce one record per five-minute recording interval. The most common REPORT option is called CIRCUIT, which produces for each interval a record containing the interval's time stamp, the number of unique to-from circuits, the number of "user" circuits (excluding beacons, etc.), number of packets on the channel, etc. The CIRCUIT output corresponding to the above log excerpt is

```
791943670,0,0,0,0,0,0,0,0,0,0,0
791943972,1,1,4,0,0,2,0,0,75,6,0
791944274,6,5,27,2,18,4,0,0,503,36,0
```

The RR option of REPORT produces shorter, CIRCUIT-like records consisting of total channel packets, non-digipeated info packets and non-digipeated <<RRn>> frames in the log file:

```
791943670,0,0,0
791943972,4,1,3
791944274,27,3,20
```

Another CIRCUIT option called RAW produces a detailed, circuit-by-circuit summary of each connection during the interval, including a sorting of packets by length, which can be used to make histograms. The RAW record for the second interval above (which contains only one circuit) is:

```
Tim Starrqp Sat Feb 04 20:26:12 1995

F Total    Total     Unique   Unique  %XD    %<32  ~64   ~128 <256 >256
  Packets  Bytes     Packets  Bytes    ON
  4        75        3        58      010    10000 090  0.0   0.0   0.0
N%MS       ~vElcRsI·7    ------ 1 frms --------   ----- Ml fr-s -------
             5           Total    NotBigi  Unique   m a l    NotDigi
                         Packets  Packets  Packets  Packets  Packets
                         Bytes    Bytes Bytes       Bytes    Bytes

                         1        1        1        4        4
                         6        6        6        75       75

                         Salrn    ua       disc     dm       rej
                         0        0        0        0        0        P

                         rnr      i        ui       fmu      poll     final
                         0        1        0        0        0        2

                         ndigi    <32      ~64      ~128     <256     >256
                         0        1        0        0        0        0
```

Further programs in the Monax25 suite perform averaging of REPORT records and totaling of log records by station heard and digipeater heard. As can be seen, the comma-delimited output of the REPORT program, in particular, can readily be processed by a spreadsheet or other software to further analyze the logged channel traffic.

4.1.2. PACKHACK

PACKHACK, written in Pascal by Bill Bradford (K7EA), is a much simpler, but still useful, version of Monax25. It analyzes a captured packet stream sent to a terminal with recording capabilities by a TNC set with MON, MCOM, MCON, MALL and MRPT all ON. When you run PACKHACK offline, it prompts you for the name of the captured data file. After filtering out offending ctrl-Z's, PACKHACK sends a table to the screen that lists, by station callsign, the total number of packets received, and the numbers of I, RR, UA, D and REJ packets. The table is not written to a file, but can be saved using a screen-capture utility. Here are the results of a recent PACKHACK analysis of a few minutes' worth of data on a 20m packet channel:

```
The PACKEIACK Chronicle for file: pkth.in

 Originating
     Station    mtal          Pacbt FYam Type
    Callsign   Packets    I      RR     UA     D  REXJ

      VE3AZD        10     6      0      0     2     0
      VP9KG7        11     9      0      0     2     0
      N4EG15         1     0      0      1     0     0
     NOMF'J-7        9     1      0      0     8     0
    VlmuD-15        19     0      0      0     0     0
       NORSU         6     2      0      4     0     0
        W9KG         8     0      0      0     4     0
       NOMFJ        15    10      0      1     2     0
       AB4UD        39     3      0      0    14     0
       w4KBs        13     0      0      0     0     0
       NBBMS         1     1      0      0     0     0
        wT6B         5     4      0      0     1     0
        wN3z         1     1      0      0     0     0
      wB6oTo         4     2      0      0     0     0
```

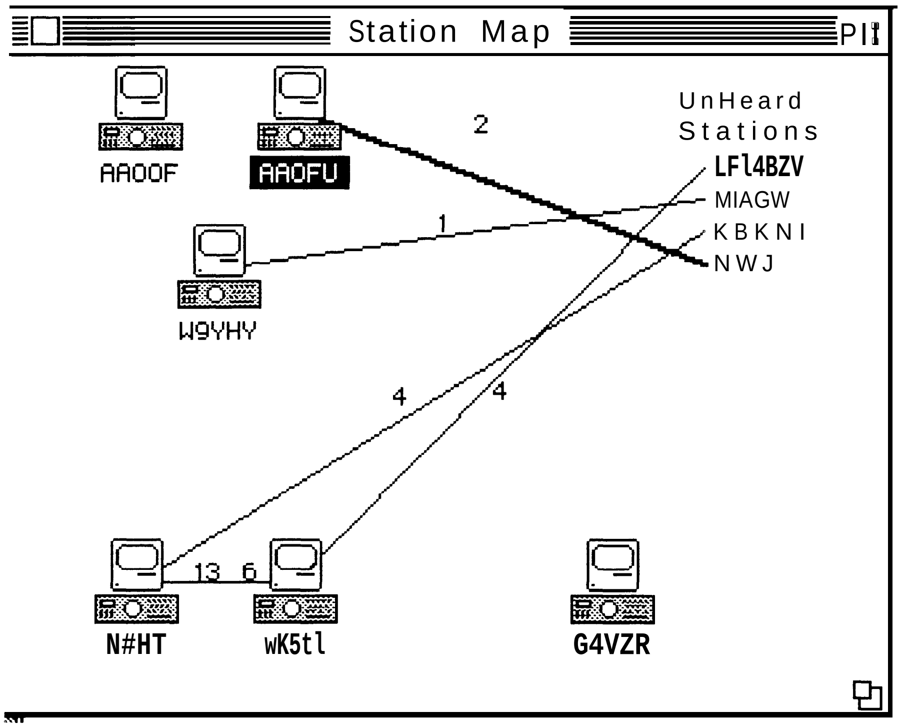
Further details on the program are in the documentation file that comes with the distribution, which can be found at ham radio bulletin boards and Internet FTP sites.

4.13. PacketTracker

PacketTracker is an AX.25 (including NET/ROM) monitoring program written by Mark Sproul (KB2ICI) that uses the Macintosh graphical user interface (GUI) to display and record network traffic on one channel in real time. Unlike Monax25, PacketTracker uses the normal TNC interface for access to channel activity detected by the TNC. Details of the program come with the distribution, which can be downloaded over the Internet from ftp .tapr.org and several bulletin boards.

The program displays information in four windows, any combination of which can be displayed at the same time. There is a MAP window, which shows individual connections between pairs of stations, a STATION LIST window, which shows all monitored stations, a STATION INFO window, which gives details of activity for particular stations, and a BAR CHART window, which displays graphs of relative "channel loading ," or "channel utilization" as functions of time.

Here's a screen capture of a PacketTracker MAP display after about five minutes of monitoring 14.103 MHz on a recent Sunday afternoon:



When a station sends a packet to another station, a thick line is drawn between them. After 30 seconds the line becomes thin, and it goes away after a user-settable time. Near these lines are written the numbers of sent and received packets. A connection with more than 10% retries shows up as a dashed line 15. Lines that extend only half way between stations indicate that one of the shown stations has timed out. If no transmission's been heard from a station, it's listed in the UnHeard column. In this example, no lines extend from AAOOF and G4VZR because they have sent beacons and are not connected to anybody. Connection maps can be saved and stations on them can be permanently attached or moved to give the display geographical significance.

The STATION LIST window displays the currently active stations, any aliases, the number of sent packets, the percentage of all monitored packets each station's packets make up and the age since last transmission. After the ages are flags that indicate the station that has most recently transmitted and whether or not it's been heard.

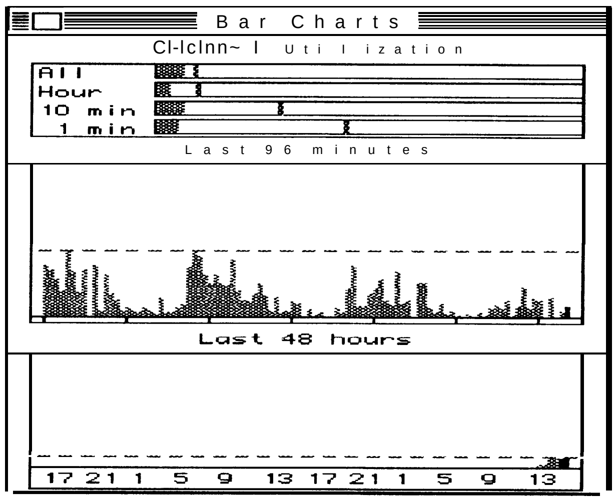
The STATION INFO window gives details for a particular station: its call, alias (if any), number of transmitted packets, number of retries (for some TNCs), age of last transmitted packet, ID string (if any) and the station the selected one is connected to (if any).

One of the most interesting and useful windows shows BAR CHARTS of "channel usage q" which is given in terms of the number of characters received from the TNC during the last minute, ten minutes, and hour, and since the program was started, along with the maximum number registered. Other graphs show the usage as a function of time during each of the last 96 minutes and during each of the last 48 hours, along with the maximum recorded usage. Full scale in each case represents ideal usage, presumably a character sent in every character-length time slot. Because of the somewhat arbitrary

'"This does not apparently happen with **KAMs** because of the way they format retried frames sent to We computer. **KB2ICI** has said he'll fix this in a future release.

scales on these graphs, they are best used to assess relative usage. Future versions of the program should perhaps make usage more precise and label axes accordingly.

Here are the bar charts for monitored traffic on 10.14887 MHz LSB on April, 23, 1995 $ from 1% 16 GMT 1[6]:



PacketTracker will also write monitored data to tab-delimited files that can be analyzed with spreadsheets or other software. These files give cumulative statistics by station, and hourly statistics on traffic from all stations.

4.2. APRS Monitoring with MacAPRS

Although it is not normally used for performance assessment, the Macintosh version of Bob Bruninga's (WB4APR) Automatic Packet Reporting System (APRS), written by Keith and Mark Sproul (WU2Z and KB2ICI)? produces a channel usage bargraph similar to PacketTracker's, and a "When Heard" table that lists the number of monitored APRS packets from each station during each hour of the day. The bargraph data and When Heard table can be saved to files for further analysis For details see the documentation that accompanies the MacAPRS distribution (at ftp.tapr.org).

4.3. NET/ROM Networks

NET/ROM network traffic can be monitored by the same means as point-to-point traffic; namely, with the AX.25 MONITOR command and its variants, and with the TCP/IP TRACE command and its variants. In addition to these, one can query NET/ROM *nodes* (after connecting to them) as to their (local) down- and uplinks by sending them the USERS command. NET/ROM nodes reachable from a particular node (and not "locked"l7) can be ascertained by sending the node in question the NODES command. Finally, certain (generally fixed) communications parameters used by a particular node can be ascertained by sending the node a PARMS or STATS command, depending on the NET/ROM software package (BPQ, TheNET, etc.) used by the node.

1[6]Although recorded near the 30m APRS frequency, this is not APRS traffic; see Section 4.2.

1[7]Locking nodes and routes defeats NET/ROM's automatic route reporting functions.

4.4. PacTOR/GTOR/CLOVER Networks

PacTOR network traffic can be monitored and recorded in the usual way at a PacTOR-equipped station in the PacTOR LISTEN mode. Link turn-arounds are usually labeled with station callsigns, which facilitates keeping links separate. Host-mode commands can be used to record data rates, turn-arounds and Huffman compression.

GTOR traffic in the (AMTOR) FEC mode can be similarly monitored, but the signal-processing required to monitor connected stations requires software dedicated to decoding the Golay codewords used for error correction in the GTOR ARQ mode 1[8]. Host-mode commands can presumably I[9] be used to simplify recording of data rates, turn-arounds 9 etc.

CLOVER's built-in monitoring facilities allow detailed monitoring and recording of performance of stations operating in the FEC mode. The close coordination of stations adapting themselves to channel conditions in the ARQ mode makes monitoring of networks operating in that mode difficult if not impossible?

Since the ARQ modes of PacTOR, GTOR and CLOVER are the ones in which truly adaptive HF communications occur (i.e., adaptation to channel conditions using *feedback),* network monitoring of ARQ traffic is a project worthy of attention from software writers. Note, however, that such monitoring is complicated by the fact that in most well designed adaptive HF networks more than one frequency will be used.

5.1. More Advanced Monitoring

Most recent work on adaptive HF networking (mainly by companies interested in military or other government sales) has been carried out under the aegis of the HF Radio Subcommittee of the National Telecommunications and Information Administration (NTIA), a part of the US Department of Commerce. The subcommittee is one of two government organizations 21 that are concerned with the development of standards for advanced HF digital communications . These standards specify various aspects of packet radio protocols designed specifically to get maximum error-free throughput from HF channels. The work ranges from the study of improvements in protocols for link establishment (including methods for over-the-air sharing of link quality measurements) to encryption of ALE signals and deciding the best way to adjust frame sizes and frame window sizes for maximum throughput in the ARQ mode when using an advanced serial-tone modem.

A typical station running such a protocol might have, in addition to transceiver and antenna, a fast PC to run the whole system, an ALE modem for linking, a device for encrypting data and a data modem with error-correction and an equalizer for transfer of encrypted data over good channels found by the ALE modem. The adaptive packet protocol is run on the PC. (Stations that are part of a new HF network used in the US Air

[1] *The v8 .O GTOR upgrade apparently allows direct monitoring.

[1 9]The GTOR mode is new, and GTOR traffic that can be monitored is relatively rare. Readers who have monitored more than one GTOR ARQ link at the same time in the host mode or otherwise are invited to send me the details.

[20]News of successful monitoring of CLOVER ARQ traffic on more than one link is welcome.

[21]The other is run by the Army at Ft. Huachuca, Ariz.

Force's Information Processing System (IPS) have this setup. IPS stations can participate in true adaptive networking, in which stations exchange channel quality and network status information, and adjust data rates, frame lengths and message routes on the fly fog highest throughput .)

Monitoring the performance of such a station during its development requires access to the asynchronous control lines between the PC and the ALE modem (two channels), the asynchronous control lines between the PC and the data modem (two channels) and the data lines (usually synchronous) between the PC and the crypt0 device (two channels). Sometimes request-to-send and clear-to-send signals (called "discretes") to or from the crypt0 device are also monitored. The monitoring thus involves at least six channels.

All this monitoring can be done from a fast PC with a big hard drive. The monitoring PC is very busy and is separate from the PC controlling the station. The monitoring PC uses serial- and parallel-I/O cards 22 mounted in its expansion slots to collect data from each channel" Since the data arrive unpredictably on the various channels? an arrangement for rapid sampling of each channel may be needed so as not to miss any data. One method is to use a ring buffer and "interrupt service routines" to poll the channels and react to any data that have arrived on them. Although this must be done rapidly, it is in practice feasible since usually only one of the modems or devices sends data at a time.

Data from each channel are usually written to their own labeled and dated file. It's almost always a good idea to have the monitoring program add *time tags* to the captured data files on a line-by-line or event-by-event basis

5.2. Statistical Analysis of Performance Data

While no fixed rules for writing performance analysis software can be written, here are some general guidelines:

Most performance analysis can be done off-line; that is, by analyzing files of data captured by a terminal program (possibly using a scripting language) or a program written specifically for data capture. In rare cases, a certain amount of on-line (real-time) analysis must be done to process incoming data before they go to a file if they arrive too fast, or there are too many to store, or their format is not suitable for immediate writing to a text file.

Analysis of a captured file often starts with *parsing* 9 which consists of line-by-line reading of the file and tests for certain characters or strings of characters that indicate events of interest ("linked to W3XYZ," "IRS," "Huffman on," "RETRIES=4," etc.). The lines, or parts of lines, containing these "flags" are then often written to a new file for further analysis or record keeping.

Numerical data in the raw data file, which are often located with a parsing test (e.g., data rates, numbers of retries, channel qualities interleaver depths, etc.) are also captured during parsing. They can be written to a tab-, space- or comma-delimited file for plotting or further analysis or analyzed during the parsing itself. (The shareware UNIX *gnuplot* program, which has been ported to PCs and Mats, can simultaneously open a data file and a file of formatting commands created by a parsing program itself; this allows one-step plotting of parsed captured data.)

[22]These cards can handle several channels simultaneously and cost two or three hundred dollars a piece.

All modern high level languages like C, C++, FORTRAN and Pascal offer extensive libraries of functions for reading or writing characters and strings, and locating or comparing particular characters and strings. These functions make parsing relatively painless.

Statistical analysis of a sample xl, ~2 ,..., XN of N random measurements (signal-to-noise ratios, LQA scores, etc.) often starts with calculations of the sample's basic statistics:

its *meun m* = *(w J ~ : I* xi, its (unbiased) *stundurd deviution*

s = *J c* "*:Ix,?l(N-1)-Nrn21(N-1/,*

and the standard deviation *(stundurd* error> of its mean s / dK'3 The sample standard deviation (and thus also the standard error of the mean) measures the spread of the data about their mean, and therefore how well the mean alone summarizes the data. The standard deviation and sample size are used to form *confidence intervuZs* for the mean, which is another way to assess how well the mean summarizes the sample.

More sophisticated counting and analysis of bit- or character errors, and numerous other measures of performance, are restricted only by the properties of the system being studied and the analyzer's imagination and programming skills. Fortunately for us hams, there are plenty of talented programmers in our ranks. I hope this article sends a few of them to their compilers, and encourages everybody else to get down and dirty with digital HF.

Acknowledgments.

I am grateful to Bob Levreault (W 1IMM) and Dave ("Dot") Willard (W 1EO) for comments on the paper that clarified several points.

Appendix: Text File Comparison.

As an example of the comparison of a received text file (generally containing errors or "missed character" symbols of one kind or another) and a stored correct file, I've written a relatively simple C-program that prompts the user for the names of received and correct files and performs such a comparison. The program counts the numbers of characters (including end-of-file characters) in each file, finds the carriage-returns in each file, and counts the number of printable, upper-case and "distinguished" (a user input) characters in the received file. A distinguished character might be a missed character symbol, as with AMTOR. The program also lists all character positions in the received file for which there is a different character at the corresponding position of the correct file. A user-prompt also allows a string-by-string comparison of the files, which helps determine the number of missmatched words; this is useful when the files don't have the same size. Here's the output of the program for files called test4 (correct) and test4r (a made-up received file). The correct file:

'"The N- 1 in place of the expected iV in these estimates offers the minor advantage that it makes the estimates unbiased! (they approach the true population statistics as N -> 00 j.

Cicero considered Syracuse the “most beautiful of Greek cities,” although
by his time it hadn’t been ruled by Greeks for four centuries.

The received version is

Cicero con-idered Syracuxe the “most b456tiful of Greek cities,” although
by his time it haxn’t been ruled by Gr--ks for four centuries.

The distinguished character was chosen to be “x” and the string-by-string option was not exercised.

The program output is

```
Correct input file is [test4].
Received input file is [test4r].

Correct/received file has 141/141 characters.

Locations of <CR>s:
Correct file has <CR> at position 74.
Correct file has <CR> at position 138,

Received file has <CR> at position 74.
Received file has <CR> at position 138.

Printable 6 upper-case character count.
Received fil.e has 114 printable and 4 upper-case character(s).

Distinguished character count:
[Distinguished character is "~"1
Received file has 2 distinguished character(s).

Whole-file analysis:
Rcvd char. 10 is 'I-"; Correct char. 10 is "s"
Rcvd char. 24 is "x"; Correct char. 24 is "s"
Rcvd char. 38 is "4"; Correct char. 38 is "e"
Rcvd char. 39 is "5"; Correct char. 39 is "a"
Rcvd char. 40 is "6"; Correct char. 40 is "u"
Rcvd char. 93 is "x"; Correct char. 93 is "d"
Rcvd char. 114 is "-"; Correct char. 114 is "e"
Rcvd char. 115 -js 'I-"; Correct char. 115 is "e"

Received file has 8 missmatch( 5.7% of characters.

Line-by-line analysis:
Rcvd line 1 has <CR> at posn 74 [Corr. char. is Oxd], 74 chars and 5 missm(es)
Rcvd line 2 has <CR> at posn 138 [Corr. char. is Oxd], 63 chars and 3 missm(es)
```

Such comparison files can be submitted to further analysis if a large number need to be summarized.