



T·A·P·R

PACKET STATUS REGISTER

Tucson Amateur Packet Radio Corporation

June 1989
Issue #35

Published by:
Tucson Amateur Packet Radio
PO Box 12925
Tucson, AZ 85732
Phone: (602) 323-1710

Editor:
Scott Loftesness W3VS
16440 Rustling Oak Court
Morgan Hill, CA 95037
CompuServe: 76703,407

In This Issue...

- President's Corner by Andy Freeborn, NØCCZ
- Annual Meeting Highlights
- PK-232 Updates
- TheNet vs Net/Rom - An Update on the Controversy
- TNC-1 Upgrade Kit Available!
- New TAPR Software Service Available
- No-Code Letter from Dave Sumner, K1ZZ
- TCP/IP New Release Notes
- New Mailer from Japan
- NK6K on BBS Design Thoughts
- DCD Mods Available
- TAPR's packetRADIO
- N3EUA's "Bits in the Basement" Column
- Info on the 8th ARRL Computer Networking Conference This Fall
- Packet Radio Related Product Announcements
- Upgrade Erratas for the DCD and TNC-1 Kits
- New PSR Editor Sought
- And Much More!

PRESIDENT'S CORNER

by Andy Freeborn, NØCCZ

Looking Ahead

The ARRL Committee to "Study a Possible Code-Free License" has made their report to the League. The report recommends the issuance of such a license. The report as a whole deserves our enthusiastic support. If we, as individuals, disagree with some element of the report we need to make that exception known. It is up to us individually to express our views to our Division Directors, Section Managers and the League. If you object to a particular facet of the report, say so. The IMPORTANT thing is that you express your view.

Lets assume for the moment that we in fact do get a code-free license and that, in time, we start acquiring the younger folks and technically oriented new members. What sort of a communications environment will these 21st century oriented individuals be joining?

There are a myriad of modes of operation in amateur radio to tweak their interest. Certainly the learning of code will be important to them. It will be needed in order to upgrade. Probably the most important mode to these folks, however, will be a mode that they can relate to immediately. In coming decades as our society becomes more and more computer literate the young folks becoming "amateur eligible" will bring with them varied backgrounds in computers and telecommunications.

We, as currently operating amateurs, need to provide them with an amateur digital communications environment which is no less sophisticated than the commercial environment to which they will have been accustomed. We need a communications environment that will solidly hold their interest for a period of time while they discover the other wonders of amateur radio. We need to drastically reduce the number of new amateurs that "drop out" after getting their licenses for lack of a mode with which they could immediately feel comfortable. I'm not saying that they should be coddled or handed something on a silver platter. What I am saying is that familiar initial communications surroundings will go a long way toward capturing our newly acquired fellow hams. Once comfortable with a mode, they will explore other facets of the hobby and become permanent fellow hams.

Many of our present day volunteer developers have been running on the leading edge of "burnout" to keep amateur radio abreast of the rest of the communications world. They have been making great personal sacrifices in the interest of amateur radio for a long time. Much of the source of future innovative technical growth must come from our newly acquired hams. The threat to future technical growth is dramatized by looking at the average age of

today's amateurs.

I recently read a no-code comment from a ham. He has been a ham for 20 years, got his license at age 30, the average age of hams at that time was thirty. Today he is 50 and the average age of hams is now near 50. Now that's pretty scary. Amateur radio needs new blood. We need new blood for continued technical growth and new blood for the preservation of the hobby.

In coming years we will need a digital environment that will permit instant global QSO's. We will need a worldwide mail message service that is fast and fully automated. At the same time we have the need to occupy and use that portion of our spectrum which has so much appeal to commercial interests.

Fortunately we have the building blocks in place today to develop the needed global systems. We have the digital R & D capabilities of TAPR and the satellite construction and launch experience of AMSAT. And there are scores of individuals and other groups doing the important independent work which is essential to system development. The work of these individuals and organizations can produce much of the needed hardware and software. The very product of their development work can attract new blood and, in turn, generate the communications environment which will require the use of the higher frequencies. And that is precisely what is needed to protect those bands for the use of future generation amateurs.

But WHOA!, first things first. We need to bring aboard those folks that can continue these programs for decades to come. Have you written to your Section Manager, your Division Director or the ARRL?

NON-TECH TOPICS

by Andy Freeborn N0CCZ

TAPR AT DAYTON A HUGE SUCCESS

Within an hour after the doors were opened on Friday at Hara Arena the 1989 TAPR booth had folks lined up to buy new kits and packet software. Until the close on Sunday the booth had a continuous stream of folks coming by. The TNC-1 Upgrade kits were the first to sell out, followed by the DCD State Machine mod kits. Nearly 1000 software diskettes were carried away. The new TCP/IP version 890421.1 went like hotcakes, accounting for 300 of the disks. The demonstration of the prototype radio modem created a lot of discussion and the descriptive brochures were picked up as if they were free \$10 bills.

UP TO OUR EYEBALLS IN DISKETTES

The newly expanded TAPR packet software service (see separate article in this issue) kept several PC clone disk drives smoking for a week before Dayton. In order to assure adequate disk supplies at Dayton and at the office 2000 disks were copied just prior to the Dayton HamVention. With Norm Miller, N0ENN and Dave Shavey, K0HOA pitching in to help we managed to get the copying job done. It would have been nice to have one of those commercial machines where you just stick in stacks of blank disks and it kicks out the finished product. Not so here however, each one was done on a PC. The night before the Dayton opening Bdale Garbee, N3EUA, keeper of the TCP/IP bits needed to make a correction on the TCP/IP Plug and Play diskettes we were to have available the next day. John Conner, WD0FHG, sat up most of the night in his hotel room with his T3200 re-doing 200 of them. Thanks Dave, Norm and John, as

popular as the software was I know that everyone appreciates it.

TAPR OFFICE HOURS REMINDER

Remember that the Tucson office is manned from 10 to 3 on Tuesdays through Fridays. Please call during those hours for orders, memberships or inquiries. (602 323-1710)

THE AMSAT JOURNAL

I recently received my first copy of the new AMSAT Journal. It's a quarterly publication edited by Joe Kasser, G3ZCZ. The first issue is a goldmine of information. I understand that it will supplement the biweekly Amateur Satellite Report. The future of packet radio networking will depend heavily on the success of AMSAT, the Microsats and future packet radio satellites. Your support of AMSAT to this end is encouraged.

NET/ROM-TheNet ISSUE

In the November 1988 issue of PSR (#33) I reported that TAPR had received an inquiry from NORD <LINK concerning the use of one of the TAPR Network Node Controllers (NNC). As a result one of the NNC development units was provided to that group. Following this there appeared several independent analyses indicating that the TheNet firmware distributed by NORD <LINK was a copy of the commercial firmware NET/ROM. At our February 1989 board meeting Ron Raikes, WA8DED, author of NET/ROM presented a paper by Thomas Allen, WA6IGY. This document by WA6IGY is titled "TheNet vs. NET/ROM Software Evaluation", dated January 1989. This detailed evaluation concludes in its final paragraph "...that TheNet is not an original development but rather a direct copy of NET/ROM." The board, then having heard from one side of the controversy, provided a copy of the WA6IGY analy-

sis to NORD><LINK and requested to hear of their side of this issue. A reply was subsequently received.

The TAPR Board of Directors having carefully considered the allegation and the response from NORD><LINK concluded that the NORD><LINK response did not effectively refute the WA6IGY allegation. TAPR has requested the return of the NNC from NORD><LINK. Both the WA6IGY analysis and the NORD><LINK response can be found elsewhere in this issue of PSR.

Letter to the Editor

Date: Fri Feb 10, 1989 8:25 am EST

From: David Sumner
To: Scott Loftesness
CC: Andy Freeborn
Dear Scott:

While they may not affect the bottom line, there are a couple of mis-statements of fact in your editorial beginning on page 13 of PSR #34 that cry out for correction — beginning with the very first sentence.

"Back in the early 80's when the FCC initially proposed a no-code license for Amateur Radio, the FCC staff thought they had the 'blessing' of the ARRL." That's a nice try on someone's part — not yours, I'm sure — at rewriting history. In fact, as reported on page 56 of September 1982 QST, "The Private Radio Bureau already knows that the ARRL Board of Directors is on record as opposing a no-code license. In fact, the Bureau told the Commissioners when it made its no-code proposal that they could expect opposition from the ARRL."

Since you're quoting Ray Kowalski, I must point out that he has given a somewhat distorted history of Radio Regulation 2735. There is indeed a provision in the current Radio Regulations for an administration to waive the Morse require-

ment for operation above 30 MHz if it so desires. However, the provision actually dates back to the 1947 Atlantic City Conference, where the frequency was 1,000 MHz. At the 1959 WARC, it was changed to 144 MHz. The U.S. proposal to WARC-79 was to leave the matter entirely to the discretion of an administration, a position that garnered strong opposition from amateurs in this country and practically no support from other administrations. In any event, the FCC no-code proposals in 1983 were not triggered by anything that happened at WARC-79; indeed, the Commission's "Experimenter Class" proposal made at that time was consistent with the Radio Regulations of 20 years earlier.

Ray Kowalski is, of course, entitled to his opinion as to whether ARRL "badly mishandled" the 220-MHz fight. Ray's main clients are land mobile interests, so he may not be entirely objective — and certainly will not share the general amateur view — when allocations issues are raised. When it comes to judging the League's performance on the 220-MHz issue — where the matter is far from settled, the fight far from over — I'm more interested in the opinions of amateurs than in the opinions of a land-mobile attorney. That the League today represents more amateurs than ever, with membership increasing by about 6% per year, suggests that the general body of amateur opinion is not the same as Ray's.

As I said at the beginning, none of this may affect the bottom line. What has happened in the past is not necessarily relevant today. As Andy said in his front-page "President's Corner" in the same issue as your editorial, the League has opened the discussion of no-code. The proponents have a fair opportunity to make their case based on present and future considerations. But it's worth it to me to keep the discussion as factual as possible, so we can be proud of the

result — however it turns out.

73,

Sincerely,

David Sumner, K1ZZ
Executive Vice President

THE 1989 TAPR ANNUAL MEETING

by Andy Freeborn, N0CCZ

The seventh annual Tucson Amateur Packet Radio membership meeting concluded a one and one-half day session on Sunday February 26th in Tucson. There was a full agenda of speakers making presentations on digital, RF, networking and satellite work in progress. On the Friday preceding the annual meeting the TAPR Board of Directors met.

The evidence of cooperation between sister organizations AMSAT and TAPR was never more apparent than at this annual meeting. Several key AMSAT officials were present and the program agenda leaned heavily toward briefings on the AMSAT Microsat satellites scheduled for launch later this year.

The fact that the first Microsats will be space based packet radio links has presented development challenges that call upon the unique talents of each organization. Several of the key players in AMSAT's Microsat development program are dual-hatted as members of both the TAPR and AMSAT Boards of Directors. TAPR has contributed \$21,300 toward development costs of the AMSAT-NA Microsat. In addition, TAPR parts procurement sources are being used to acquire many of the needed electronic components.

The eight and one half hour Saturday session started at 0900. TAPR board members Dr. Tom Clark, W3IWI and Harold Price, NK6K

spoke on Microsat topics. Both Harold and Tom have been active for many years in AMSAT satellite development efforts. TAPR member Jon Bloom, KE3Z, described the Microsat Power Module being developed in the ARRL labs. AMSAT's VP for Engineering and Microsat chief Jan King, W3GEY, displayed and described a full scale Microsat satellite. Other key AMSAT players present were AMSAT-NA President Doug Loughmiller, KO5I, and Brazil AMSAT President Dr. Junior DeCastro, PY2BJO. Lyle Johnson, WA7GXD, long time former TAPR President and a key Microsat developer was out of the country on business and unable to attend. Also unable to attend was Dr. Bob McGwier, N4HY, a former TAPR board member, present AMSAT board member and key Microsat developer.

The wide variety of other topics presented gave a good representation of the interests and activities of TAPR members. These presentations covered TCP/IP, TexNet, HF BBS Networks, no-code license, 1200 MHz transverter, microwave Ethernet, the K3MC I/O board, the 56 kbps modem, modem demodulation experiments, TAPR hardware projects, 10 GHz EME and recent ARRL actions of interest.

The Sunday session of 3 1/2 hours was devoted exclusively to discussion of a no-code license. It was a lively and constructive discussion and TAPR no-code committee chairman Harold Price accumulated a great deal of valuable input from the membership.

Re-elected TAPR President Andy Freeborn, N0CCZ, announced the results of balloting for the five vacant Board of Director seats. Re-elected to the board were Steve Goode, K9NG, Eric Gustafson, N7CL, and Lyle Johnson, WA7GXD. New members to the board are Bdale Garbee, N3EUA, and Franklin Antonio, N6NKF. In addition to Free-

born the new officers are Pete Eaton, WB9FLW as Executive Vice President, Dave Toth, VE3GYQ as Secretary and Bdale Garbee, N3EUA, as Treasurer.

Paul Williamson, KB5MU, from the SANDPAC Newsletter prepared a comprehensive "Blow-by-Blow Report of the 1989 TAPR Annual Meeting". This excellent 13 page summary of each talk can be obtained from the TAPR office by sending an SASE with 3 units of postage to the TAPR office (TAPR, Box 12925, Tucson AZ, 85732).

1988 FINANCIAL REPORT

The 1988 financial report was presented to the membership at the February Annual Meeting in Tucson. Following is a summary of the years operating activity.

INCOME	
Dues	4,916.00
Sales	19,131.36
Interest	3,719.86
Royalties	33,953.00
Miscellaneous	193.26
Total Income:	61,913.48

EXPENSES	
Cost of sales	780.81
Admin. & Operating	17,840.79
Printing & Publication	5,260.67
Research & Development	21,960.34
Taxes, Ins, Deprec., Misc.	6,434.86
Total Expenses:	52,277.47

Operating Income 9,636.01

8th NETWORKING CONFERENCE TO BE AT AIR FORCE ACADEMY

This years Computer Networking Conference (the 8th one) is being held at the Air Force Academy in Colorado Springs CO on Saturday and Sunday the 7th and 8th of October 1989. Prior conferences have been held on the west coast or in the east. With Colorado Springs being more centrally located the event this

year should be more convenient for a larger number of people to attend.

Don't let the title of the conference fool you, the subject matter is all PACKET RADIO.

The hosts this year are TAPR, Academy Amateur Radio Club, USAFA Cadet Radio Club, Rocky Mountain Packet Radio Assoc. and ARRL.

The Saturday session at the Academy will include prominent amateurs who are doing packet radio development work at the leading edge of technology. They'll be speaking about their current efforts in the areas of packet satellites (PACSAT), networking (TCP/IP, TexNet, ROSE), the new version of AX.25, digital signal processing (DSP), high speed packet (10MB/sec), new packet hardware and packet software developments and many other fascinating development efforts now in progress. Luncheon on Saturday will be at the AFA Officers Club transportation to and from the conference area will be provided.

On Saturday evening there will be an opportunity to get acquainted at a special (and informal) attitude adjustment session at the conference hotel. A full size Taco buffet (\$2 per pass through the buffet) and cash bar will be available. Hours 6:30 to 10:00. This will be a great opportunity to eyeball with some of the folks that you have been packeting with and to ragchew with some of the conference speakers.

There will be two concurrent sessions at the conference hotel on Sunday. The ARRL Digital Committee will be in open session and amateurs are welcome to sit in. At the same time the Rocky Mountain Packet Radio Association will be hosting their annual Packetfest in a separate meeting room.

(continued on page 22)

Design Thoughts on the Pacsat BBS, and How It Differs From a Ground-based BBS.

by Harold E. Price NK6K

May 27, 1989

Pacsat is a generic term used to describe a digital store-and-forward satellite mission in the Amateur Radio Service. Two of four Microsats and one of two UoSATs scheduled for launch in November of 1989 will have a primary Pacsat mission. This article contains some of the ideas I've been inserting into protocol design discussion in the Pacsat world. I've volunteered to write the Microsat BBS, G0/K8KA will be writing the UoSAT BBS. Since both will use the same spacecraft operating system and application development tools, and both will orbiting the same planet, it seems only natural to work on a common set of protocols and procedures.

Although it has been agreed that Pacsats will use the AX.25 frame as the basic link layer protocol, either in the full AX.25 connected mode, or as UI-frame datagrams, exactly what goes in these frames has not yet been decided.

Here are some of the types of information that a Pacsat will deal with:

- 1) Forwarded mail messages. These are messages that are not destined for Pacsat as an endpoint, but are in transit between forwarding gateway stations.
- 2) Personal electronic mail messages. These are messages to and from individuals who are using the satellite as a BBS; entered either directly by a human-run connection, or by using a program that pre-formats messages for fast transmission. These messages use Pacsat as

an endpoint. It can be argued that all mail should be forwarded mail, that no one use Pacsat as a direct BBS. There are three reasons to permit direct access:

- a) For international access in remote areas that do not have a terrestrial infrastructure in place.
 - b) For emergency access by minimal ground stations.
 - c) To permit large numbers of people to have a direct hands-on experience with satellite communications.
- 3) Realtime Telemetry. These are current spacecraft telemetry values, such as solar array power, internal temperature, etc.
 - 4) Stored Telemetry. This is a file of one or more telemetry values stored over time, for example, the output of the solar arrays once a second over the last orbit.
 - 5) Bulletins. These are items of general interest, orbit predictions, AMSAT News, Gateway, etc.

This article discusses thoughts on how to handle the BBS to BBS aspects of Pacsat. It should be noted that no final decision has been made on how this will be done, these are my current thoughts on the subject.

First, let me split the discussion in two parts, access method and forwarding method. The access method is how one BBS transfers a message to a second BBS, or in this case, to Pacsat. The current access method is that you connect up like a user, and use the same commands a user would, with some additional logon handshaking enclosed in square brackets. This information is usually the type of BBS, RLI, MBL, GYQ, etc., the version, and what special features are implemented.

Forwarding method is how you decide what to send where, and how the next BBS knows what to do with it next.

The majority of my comments are in the first category, access method. The following is not meant to be taken as a negative comment on the WORLI-descended forwarding access method. The current scheme is as it is for several good and valid reasons, I'll not spend time discussing them here. But the current way of doing things grew from a much different environment with far different constraints than a Pacsat will encounter.

As I see it, the over-riding attributes (design drivers) of Pacsat are:

- 1) Limited access time per pass. Very very very limited.
- 2) Full duplex
- 3) No need to accommodate users and BBSes with the same interface.
- 4) Not as limited a file storage capacity as some folks think.
- 5) It is in view of far more potential lids, either malevolent or just uninformed.

These are sorted highest to lowest impact. Let's discuss them in reverse order.

In view of far more potential lids...

If someone wants to make trouble on a BBS, he's usually limited to his local area. If someone gets it in his head that the Klingons killed his brother on Rigel-7, and wants to read and delete all mail to Klingons, he can cause far more trouble on Pacsat than he can on the ground. We needn't worry too much about this sort of thing; 90% of computer security is common sense, and getting the other 10% is on an asymptotic cost curve. But there's no sense in leaving the door unlocked, either. The simplest example is users deletion of mail.

There are two reasons to delete mail. The first is so a user can avoid read-

ing mail more than once. He reads it and deletes it, and need not be concerned about re-encountering it. The other reason is to help the sysop purge old messages.

On Pacsat, we want an easy way for users to get personal mail they haven't seen, while at the same time keeping them from reading mail they have seen. This is usually done by an "owner has read it" flag, or a list-since-last-logon command. These are prone to attack by Klinton haters who read your mail for you. (I'm not making that up, by the way, it's happened before in my local area). The best way is to give the user a "display mine after message N", and he (or his software) remembers N.

Pacsats can't afford a sysop. This will come up time and again. One reason is that sysoping a world-wide BBS is a big job, doing 3 is worse. But that's not the limiting factor, link access time is. For a sysop to make an informed guess as to which message is old and needs deleted, he has to read it. Them. All of them. Downlink time spent on administrative functions is overhead, and we'd like to reduce that as much as possible. To avoid then need of a sysop for disk maintenance, when the ram disc is full, the oldest message gets tossed. Exceptions: A control op can make a message stick, for schedules and the like, and a control op can delete messages that are illegal. That last item requires reams of paper, and it isn't discussed further here.

The way this will probably be handled is that each message will have an expiration date. Once RAM is full, messages that have expired are automatically purged. It takes far less time to put a valid expiration date on a message at upload that it does to periodically review messages. In addition to "purge after N days", options would exist to purge as needed, and purge never. Only control stations or bulletin

sources can set specific expirations dates, regular users get a default, probably purge as needed.

So, one design item that comes from the different environment is that a user can't delete messages, even his own. Not that I want to make a big deal about deleting messages, its just an example of things which are taken for granted on the ground but are different in orbit.

Not as limited a file storage capacity as some folks think

There will be 8 Mb of data storage on microsat, (6mb + 2mb bank switched), and 4.25 MB on UoSAT, for a total of 20.25 MB. We don't need to dismiss out of hand schemes that require a 100k control file. Not that anything I'm proposing does, I've just noticed a tendency for people to say "Of course, you can't have a 80k forwarding control file on Pacsat". A 80k control file is just 1 or 2 percent of a Pacsat, and that's not a problem. The problem is how often the overhead files need updated, a link access time problem.

"No need to accommodate users and BBSes with the same interface."

Pacsat has a multiple endpoint AX.25. That means that each task in Pacsat can own one or more call-signs. That allows a ground user to talk to a program by "connecting" to the program he wants to talk to. This simplifies things greatly, as we don't have to have one big menu program, e.g.,

Select 1:

- 1) Update housekeeping tables
- 2) BBS
- 3) Upload new program
- 4) Change telemetry cycle
- 5) reboot

that greets you when you connect, and which would then re-direct the bytes to the target task.

In real X.25, this function is taken care of by circuit numbers, in TCP it's taken care of by ports. Since Pacsat is LAPB based, we'll use multiple addresses as a de-multiplexing function, e.g.,

msata-0 User BBS port
msata-1 Housekeeping
msata-2 Loader
msata-3 debug (read-write memory)
msata-4 Forwarding

Immediate implications: No need to have forwarding be limited to a protocol that can sort humans from BBSes, which means no need to have the forwarding protocol pretend it is a user. It is this pretending to be a user that makes BBSes harder to implement, and makes them inefficient users of channel time. More about that last bit later.

Note that there is no need to limit the programs to the same call with a different ssid, both can be changed. There is no need to have the call sign be a real call anyway, as stations in the space service don't have the same ID rules.

Full duplex

What a waste it would be if the Pacsat BBS worked the same way that ground BBSes work; all the traffic goes from station A to Station B, then the flow reverses, Station B sends messages to Station A. Pacsat is full duplex, we should strive to make use of that, the limited access time demands it.

Limited access time. Very very limited.

The two Microsats and one Uosat will be able to transmit, if they are 60% efficient, a total of about 900 data bytes per second, about 40 minutes a day. That's 2.1M bytes a day. Even at 18% efficient there is still 600k bytes a day, from anywhere on the planet to anywhere

else on the planet. That's a lot more bytes than we get now using HF.

But we won't see that if we send data the same way ground BBSes do it now. Take your current RLI-derived bbs, and start a stopwatch. Turn your radio off after 10 minutes. Turn it on after 90, then off again in 10. Come back in 12 hours. Repeat. Because any message in transit at the time you cut off the link is lost and must be restarted, at best you'll have wasted a non-trivial amount of channel time. At worst, you'll get stuck on one message that takes longer than 10 minutes to send, and that's the last useful thing your BBS will ever do until you chop the message up by hand..

The amount of waste depends on average file size. The size of a stuck file depends on conditions, are you using an omni or are you tracking, the orientation of the satellite at that time, and the number of other users in a 4000km range circle. It also depends on the spacecraft, 1200 baud for Microsat, 9600 baud for UoSAT.

I think Pacsat's place in the scheme of things is the international (or coast to coast for big countries) transfer of large messages, the kind HF sysops blanch at. For example, some of the larger bulletins and newsletters are about 20k, some of the manuals for the spacecraft software are as much as 160k. These are not the kind of things you would want to forward on HF, but they would be fine for Pacsat gateway to gateway, and for a higher-speed local area VHF/UHF link.

Currently, even 8k is large for HF. On an average day, 8k might be as much as 10% of a Microsat pass. If you had to restart a message from the top, 10% of a pass would be wasted.

It is, therefore, imperative to have the ability to continue a message transmission on a subsequent pass,

which would require a special forwarding protocol.

Review

In way of review, what attributes of the current BBS access method are undesirable on Pacsat?

- 1) All partial transmissions must be started from scratch. Partial transmissions are a fact of life in the Pacsat low earth orbit, a pass has a finite length. At best, "no restart" results is very inefficient use of a sparse resource. At worst, some messages, and in particular the kind of messages that Pacsat would be best for, will get stuck in the pipe.
- 2) It is half duplex, if not de jure, de facto. Adding full duplex to current implementations would be a major pain. Staying with half duplex is wasting 1/2 of the capacity, right off the top.
- 3) It has a few kludge handshakes, made necessary because you don't know if you're talking to a user or another BBS, or what kind of BBS.
- 4) There are many BBS designs, with different interoperability problems, some solved by the [] kludge, some not. As example of not, headers could be used to detect routing loops if they were universally useful. This is fixed in practice by an attentive sysop, something we'd like to avoid on Pacsat.

PACSAT forwarding BBSes

I'm not sure what most people picture in their head when they think of a Pacsat-capable forwarding BBS system. Here are my views.

- 1) The human-in-the-loop system. This is a guy who sets his alarm clock for the next pass, comes in, re-configures his station for Pacsat, steers the antennas, the radios, and the software by hand. This is fine for the individual, occasional user. But if this user

wants to be a real forwarding node, he'll get real tired of this real quick, and will either upgrade or drop out. Either way, we don't need to optimize for him, and he probably won't carry much traffic anyway.

- 2) The simplest "automated" system, operationally, is to have standard software and dedicated Pacsat serial port, with dedicated radios. You have an omni antenna, and treat Pacsat like any other BBS. This won't work though, since you have to know when to try to forward to Pacsat. Some additional software or modifications to the BBS are required.
- 3) Same as 2), but with added software mods to know when to forward. This is better, but unless you steer your uplink frequency, you'll only get part of the orbit. More mods are required.
- 4) Same as 2), but now your BBS has to know how to steer your radio, or even better, also your antennas. I can't see each BBS author supplying code for each radio/antenna rotor, so we can now assume that the system has the capability of running more than one program concurrently, desqview, doubledos, etc. One program is doing the steering, and one the bbsing.

So now we arrive at what is the more likely average forwarding system, one that is concurrent-program capable. We've also made the leap to the concept of the other program running some of the show.

A second leap, a likely implementation.

It is likely that one of the other programs will be the Pacsat access program. It requires a far smaller mod to existing BBS programs; it requires only that they be able to forward into and out of a file.

Some BBS systems already have the capability of exporting and importing messages into a file, for exchange

with another program. The pacsat forwarding program would take these files and exchange them with a forwarding program on Pacsat.

This other program would also contain data compression, something else that would be very nice to have sooner than later. Placing it here relieves the BBS authors from having to add that into already burgeoning .EXE files.

So what's it get us?

Splitting functions between two programs gives us an increase on both software development efficiency and link efficiency.

It moves the requirement of getting agreement on the myriad other details, such as exact spacecraft access protocols, compression schemes, restart, etc. from the set of all BBS authors to a smaller set of spacecraft interface authors.

The new code can be written once, and source distributed.

It also allows for simpler implementation of different access protocols. Some examples are a broadcast protocol, or the DX-list style access that K8KA discussed in the 7th ARRL Networking conference proceedings, where Pacsat says "Yes, you are connected, move to uplink 2 and send", or "all uplinks in use, send priority traffic now, or other traffic at 10 frames per minute." It decouples the BBS from the Pacsat forwarding, and it doesn't cost anything more than the ability to forward into a file.

Although we like to say that Pacsat is just a floating BBS, there are some differences. To get the most out of a Pacsat, we should choose access protocols wisely.

Renewals?

Check your address label for the expiration date of your TAPR membership. Please RENEW!!!

PRODUCTS AVAILABLE FROM TAPR

HARDWARE

Hardware kits that are currently available from TAPR are shown below. All prices include S & H.

PSK Modem	\$110.00
K9NG 9600 Baud Modem	25.00
TNC2 Tuning Indicator	25.00
XR2211 DCD Mod.	11.00
State Machine DCD Mod.	17.50
TNC 1 Upgrade to TNC 2	59.00
Memory Kit for TNC 1 Mod.	20.00
(When purchased w/TNC 1 Mod - includes 32k RAM and 1.1.6 w/ KISS EPROM)	

FIRMWARE

The TNC2 software version 1.1.6 is available with KISS. If you have been using version 1.1.4 or 1.1.5 with the 32k RAM you will be able to upgrade directly to 1.1.6. For those still using 1.1.3 it will be necessary to install the 32k RAM at the same time that you upgrade to 1.1.6. Installation instructions are provided with the 32k RAM.

TAPR will program your EPROMs for \$2 per TNC-worth plus a pre-paid return mailer. If you choose to buy EPROMs from TAPR we will include the mailer and postage in the purchase price of the blank EPROM.

Prices as follows:

32k RAM (includes update doc)	\$20
Blank EPROM (27C256)	
(add \$2 for programming)	\$10
Blank EPROM (2764)	
(add \$2 for programming)	
(may be 27C64 if available)	\$5

PROGRAMMED EPROMS

TNC-2 WABDED (27C256)
TNC-1 WA8DED (2 x 2764)
TNC-1 KISS (2764)
TNC-2 1.1.6 w/KISS (27C256)

TAPR's NEW AND EXPANDED SOFTWARE SERVICE

Please order by Disk Number. Including an addressed mailing label will help.

Disk

1. APLINK - W5SMM - Runs MBO & BBS
2. BB - AA4RE - A multiconnect MB
3. C-BBS - K3RL/AG3F - MB w/ sources
4. EZPAC11 - M. Imel - NTS formatter
5. MONAX - NK6K/WB6YMH - Gathering system stats
6. Packet S/W - WB6UUT - for PK 87,88,232
7. PBBS Lists - W9ZRX - Master PBBS lists
8. R95 - WD5IVD - Binary conversion utility
9. ROSESERV - KA2BQE - BB and server for ROSE
10. ROSE Switch - W2VY - The Rose executibles
- 11/11a TCP/IP Plug & Play - KA9Q - ver 890421.1 (2 DISKS)
- 12/12a TCP/IP Sources - KA9Q - ver 890421.1 (2 DISKS)
13. TNC-1 Source code - TAPR - TNC-1 sources
14. TNC-2 Software Notes - N2WX - 1.1.0 thru 1.1.6
15. WA7MBL Mailbox (BBS) - ver 5.12
16. WORLI Mailbox (BBS) - ver 10.04
17. YAPP - WA7MBL - terminal program ver 2.0
18. INTRO TO TCP/IP - Much info on TCP/IP (2 Disks)

Diskettes are \$2 each including diskette, mailer and postage. Please do not send blank diskettes, mailers or postage. For orders outside North America please add \$2 for airmail delivery.

If later versions than those shown are available they will be substituted.

Send your orders for any of these products to TAPR, PO Box 12925, Tucson, AZ 85732.

TheNet vs. NET/ROM

At the annual TAPR Board of Directors meeting in Tucson in February 1989 Ron Raikes, WA8DED, author of NET/ROM, formally presented his argument that the TheNet firmware was a copy of NET/ROM. Ron substantiated his argument with an independent analysis by Thomas Allen, WA6IGY, titled "TheNet vs. NET/ROM Software Evaluation". Ron's concern centered on the fact that TAPR had provided one of its Network Node Controller (NNC) software development systems to NORD<>LINK, producers of TheNet.

The TAPR Board, having heard from one side of the controversy, felt compelled to hear from the opposite side before drawing conclusions. Consequently the WA6IGY analysis was sent to the NORD<>LINK group with a request that they provide us with their comments concerning the allegations.

A response from the NORD<>LINK group has been received.

The TAPR Board of Directors have evaluated the response from NORD<>LINK and have concluded that it does not adequately address the issues raised. As a result the NORD<>LINK group has been requested to return the NNC to TAPR.

Andy Freeborn N0CCZ
President
TAPR

The NORD<>LINK response follows:

— Start of statement —

1. We state,

- a year passed by since the first release of TheNet

- neither Software 2000 Inc., nor Ron Raikes WA8DED, nor Mike Bush W6IXU, send us (NORD<>LINK, DF2AU, DC4OX or anybody else involved in TheNet) directly or thru third parties any letter, note or whatsoever telling us to stop distribution or accusing us of copyright infringement. All we read were slander, libel and false accusations on CompuServe and thru packet Radio.

2. We state that for more than a year Ron Raikes WA8DED and Mike Bush W6IXU are using a language we don't share:

#: 75026 S9/Packet Radio
08-May-88 16:09:01
Sb: NET/ROM Ripoff!
Fm: Mike Busch [W6IXU]
76337,727
To: All

Folks, let me make myself perfectly clear. This is unabashed thievery. THENET is a ripoff of NET/ROM plain and simple. Detlef J. Schmidt DK4EG and his German NORD<>LINK cohorts are thieves.

I am told that these are the same bums that ripped off TAPR's TNC-2 circuitry and firmware in Europe.

This is just one example of many following but we don't want to talk that way.

3. We state that for more than one year Ron Raikes WA8DED and Mike Bush W6IXU are telling lies and slander.

Here is just one example of many others from the above mentioned letter.

For all intents and purposes, the Germans simply removed our copyright notice, relocated a block of constants to the front of the ROM, put their name on

it, and started distributing it.

To show that this is not true we immediately published the source. One of the latest accusations is that TheBox by DF3AV would be a copy of W6IXU's Multibox. TheBox is public domain as source code and the very first look would show that S2000 lies. We can not give details because we never had a copy of the Multibox. The box version mentioned in one of the telexes between DJ3UW and WA8DED was NOT the Multibox. In fact we never had a copy of the object or the sources of that software and DJ3UW is not linked to us in any way. He just sold the NETROMs to us. He is not active in packetradio any more for years now.

4. We never denied that TheNet is a clone of NETROM. Just the opposite is true. The following is from one of the first public answers, about one year ago (Original by DC4OX, translation by KE6MN).

#: 75225 S9/Packet Radio
12-May-88 05:31:07
Sb: #75026-#NET/ROM Ripoff!
Fm: Don Moe KE6MN/DJOHC
72407,1054
To: Mike Busch [W6IXU]
76337,727

And how is the 100% functional compatibility verified? Correct, one attempts to approach the original as closely as possible, and then generate better code where possible. If Ron Raikes has put so much time into looking at the code for TheNet, as Mike Bush wrote, then he must have seen that we used a different compiler and another optimizer. (Naturally we bought the compiler officially and didn't swipe it.)

There were more and more detailed letters in our mailboxes. We don't know how much came over to you. In one letter DC4OX stated that during the development of TheFirmware he had achieved a byte by byte

identical code of WA8DEDs firmware. By the way, we also started with the firmware, not with TheNet, although WA8DED claims it different. Again: we do not want to prove every lie of S2000. It just takes time and doesn't help at all.

5. If you have the sources of TheNet and NETROM everybody can tell within minutes (software professional or not) if the sources are similar or not. Within a day everybody could tell if they are identical or not. If this procedure takes more than half a year until being published we get suspicious.

Because we published our source first nobody can tell how much the S2000 sources changed since then. A remark: during the cloning process we found everything from real bugs to unused variables and potential errors (when using a different compiler) in WA8DEDs code. So somebody has to be wrong.

Again: neither S2000 nor WA8DEO nor W6IXU ever asked us what and how we did it.

6. Everybody familiar with software - even non-professionals - can see at first glance at our well documented sources (well documented even if it were professional software) that it would have been a very easy task for the authors to change everything in such a way that similarities with NETROM would only be visible after very extensive investigations. If the authors had wanted it that way...

If the only intention would have been to enable copying of NETROM it would have been sufficient to publish the call encryption routines and all the locations where the default data are located. This would also have enabled legal owners of NETROM to change the callsign if needed to do so.

We feel sorry for WA6IGY because he invested so much time to find

differences which we would have told him on request.

But we want to complete some of his assertions:

8) The IDENT command was not simply renamed to INFO. INFO has a different meaning and does different things. Also the password is initialized from the EPROM (opposite to NETROM where after a power failure it is lost).

9) The hand optimized routines are NOT identical to NETROM. Also the TheNet sources include the portable C routines too, which were the basis for our optimization. Not regarding our "better hands" you can easily see that we used a different compiler.

13) TheNet supports the TNC220 instead.

14) We don't know the NETROM source but from our previous experiences with WA8DED's products (that came here as source code) we think that our version is easier to understand because of its very extensive comments. Naturally you have to know German. Some of Ron's programs were very hard to read, even if someone was fluent in English.

Unfortunately Thomas Allen had no Q/C Compiler. Now we sell the compiler complete with its source and the library source at a real HAM price (together with the optimizer, also complete with source, it is about two NETROMs). The money goes 100% and directly to the author. And Thomas Allen forgets something in his statement. Although Q/C is a very poor optimizing compiler (he states this as "not optimizing", please look at the compiler sources if you do not believe) two very extensive optimizers were used. One of them was common to S2000 and us (AO by Logical Systems, replaces JMP with JR where possible and replaces some CALLs with RSTs)

and the more extensive homemade. These optimizers take some 30% out of the code. So his conclusions on the way of reverse engineering are a bit misleading.

7. "considerably changed and improved" or "only minor changes" ?

S2000 says that TheNet were NETROM 1.3 with only minor changes. S2000 is considering the common source lines, we consider the number of bugs and features.

While reading the following statements please keep in mind that S2000 said that NETROM development has ceased with release 1.3. Because of limited space in the EPROM there is no room for improvements. In other words: they made all the money they could have and now you have to live with it.

We know that NETROM 1.3 has some real big bugs. As an example there is an erroneous handling of received I-frames with pollbit set while in reject state. NETROM will ignore these frames causing the system to hang if used with some AX.25 implementations. This is fixed in the current release of TheNet. Just a minor change and easy to find by disassembling the EPROM?

If a user goes uplink or downlink thru Level 1-2 digipeaters you will not see this in NETROM but in TheNet.

If you need to coldstart the TNC without loosing the Sysop-Remote-feature you have to take TheNet, NETROM cannot do this.

If you need some remote control for e.g. antennas, a PA, frequency selection you have to take TheNet, NETROM cannot do this.

If your backbone runs full duplex (as most backbones do here in DL) you have to take TheNet, NETROM cannot do this.

If your modem needs flags while not transmitting data you have to take TheNet, NETROM cannot do this.

If you have a duplex digi which has to send flags as a busy signal because a single tone will be ignored by some modems you have to take TheNet, NETROM cannot do this.

If you want conversational mode with round table chatting you have to take TheNet, NETROM cannot do this.

If you want your ATARI-ST or IBM-PC attached to the RS232 inter-node communication for use as a powerful host computer you have to take TheNet, NETROM cannot do this.

We do not want to state every single improvement. Here is not enough room for that but the above mentioned should make clear why we continue to say "considerably changed and improved" and why we do not agree with S2000's cynical statement "networking software for the TNC2 has come to an end with version NETROM 1.3". We will continue to improve TheNet and all amateurs are invited to join us. Future node controllers will run the same software as the good old TNC2 (which by no means has come to its end).

8. We stayed away from all the discussions with S2000 as much as possible because we are in no way delighted by slander, libel and false accusations. And that seems to be their favorite way of "discussion". This will not only hamper the reputation of some single hams but the whole community of amateur radio. We still see amateur radio as a way to do technical studies in cooperation with all other hams.

— End of statement —

[Editor's Note: WA6IGY's analysis of the controversy follows. It is printed here to provide additional insight into this issue.]

TheNet vs. NET/ROM Software Evaluation

by Thomas Allen WA6IGY
January 1989

After a careful, independent, detailed analysis of these two popular network communication packages, I have reached the conclusion that NordLink's TheNet version 1.0/1.1 is a copy of NET/ROM Version 1.3 and could only have been created by disassembling the object program from a NET/ROM 27C256 EPROM and then reconstructing equivalent source programs in the C language and assembly language. The NET/ROM design was copied in its entirety.

During the late summer of 1988, I obtained the NET/ROM program from the author and TheNet from local sources. I also reviewed a significant number of messages on CompuServe's Hamnet related to the NET/ROM dispute. The controversy on CompuServe of whether TheNet was a copy of Software 2000's NET/ROM was clouded because only the hex files of both programs were available. The source program of NET/ROM is copyrighted and was not available to those who had already done their own evaluations and based their decisions on comparing the hex files only.

Because of the disassembly and de-compilation technique presumably used to clone NET/ROM and the lack of identical variable and data names in the source programs, a strictly programmatic comparison of the source programs proved infeasible. However, during a very tedious manual review, I was able to devise ways of automating the process somewhat.

My findings show that the source code from both programs is the same, statement by statement, with only variable, data, and structure

names changing. Names would not have been present in the executable code (from the EPROM) and could not have been expected to survive the disassembly and de-compilation process used.

From the manual examination of the source files, I could find no evidence that TheNet is an independent design from specifications or a complete rewrite. Though repeatedly claimed by the "authors", TheNet is NOT a superset of NET/ROM, and in fact, is NOT an original work by any stretch of the imagination.

NET/ROM vs. TheNet, a Software Comparison

During the late summer of 1988, I obtained the source files of NET/ROM 1.3 from the author and the source files of TheNet version 1.0/1.1 from local sources in order to do an independent comparison in light of claims by Software 2000 of copyright infringement by the Northern German Packet Group, NORD<>LINK. This document reports my findings.

NET/ROM is a firmware replacement which converts a regulation TNC-2 to a packet radio network node controller. It was written by Ronald Raikes, WA8DED. Although I am a packet enthusiast, I have no connection with Software 2000 nor any proprietary interest in NET/ROM. My interest in doing this evaluation was purely technical.

First Things First:

I started this activity by copying all the NET/ROM and TheNet source files to my hard disk. After removing all tabs, the files were printed on a laser printer and separated into two 3-ring binders. After considerable time reviewing the 2-inch stack of listings wondering how to attack the project, I discovered a curious and consistent similarity. Some routines in one set appeared to have a counterpart routine in the other

binder; they were of similar lengths, had the same number of formal calling parameters, same number of local variables, and virtually identical form of construction.

I decided to create a cross-reference table of routine names and their associated file names so I could keep track of this manual progress through the files. I "visited" every routine in the NET/ROM binder and copied the procedure name, the file name, and the number of parameters into a text file. After sorting on the column of procedure names, I printed the file to use as a worksheet.

I began to search through the set of TheNet files to find some correlation with the NET/ROM routines I had already cataloged. By narrowing each search pass to just those files dealing with a single protocol layer (starting with layer 7), the table was filled in rather quickly. Nevertheless, this effort took over two weeks of part-time work. For every NET/ROM routine, there was a matching NordLink routine, but I had four TheNet routines left over which had no match in NET/ROM. The result of this effort was a four page reference of all routine and file names and number of parameters.

I compared every pair of routines visually, line by line. When I ran my index fingers down each page, the same pattern recurred; an IF followed by an assignment, followed by a procedure call, followed by a pointer increment, followed by a call, etc., ad infinitum. Every called procedure in TheNet could be cross-referenced to the matching NET/ROM routine I had cataloged. When NET/ROM called a C routine, so did TheNet. When assembly language was called, so did TheNet. I became progressively more frustrated at the slim prospect of doing this by some automated means, but I continued with this painstaking process to the bitter end.

Findings:

1 There are 234 NET/ROM routines in version 1.3. I define "routine" as an executable code segment named as public (global), which includes all C functions and entry points to assembly language code.

2 One routine in NET/ROM, `crif` (file `LAYER7CN.C`), is not referenced therein and has no complement in TheNet. This widowed code was probably an oversight from a previous release of the firmware.

3 One routine in NET/ROM, `staind` (file `TNC2N.MAC`), is not referenced. The matching routine is referenced and used in TheNet as `STAled` (file `TNL1.MAC`).

4 Of the remaining 232 routines in NET/ROM, all are duplicated in TheNet with identical numbers and types of passed parameters. In cases where there are two or more parameters in calling arguments, the order has been consistently reversed in TheNet. Reversing the order of the parameters was no doubt due to an individual's preference.

5 In every TheNet C function, an identical number and type of auto variables are allocated on the stack in the same order as they are in the corresponding NET/ROM routine.

6 All structures in NET/ROM having preset data have an identical analogue in TheNet including order and type of data initialized. This includes all character strings and procedure jump address tables.

7 TheNet routines `l2init` (in `L2E.c`), `l3init` (in `TNL3.C`), and `inivar` (in `TNL7A.C`) differ from the corresponding NET/ROM routines only in that a single statement has been deleted to remove call sign encryption. `l2init` of TheNet has one additional procedure call related to cold-booting.

8 Full duplex was later added to

TheNet routine `hstcmd` (in `TNL7C.C`). This added a 20-line case 'F' to an existing switch statement and comprised three if statements, six function calls, and two assignments. In assembly language, 16 bytes were required to complete this modification, including 3 lines in routine `kickbx` (in `TNL1.MAC`) and 11 lines of a new module, `pushtx` (in `TNL7B.C`). The `IDENT` command was renamed to `INFO` and the `sysop's` password was initialized to a different string, both minor changes.

9 In NET/ROM layer 2, nine interrupt service routines dealing with low level I/O and buffer allocation and de-allocation were manually recoded by the author to ensure an adequate processing margin at 9600 bps. These functions were originally written in C for the AX.25 Level 2 user firmware for the TNC-2. An assembly language source file, created with a Q/C compiler option, was used as a starting point. It was then hand-optimized and assembled. This optimized set of assembly language functions is identical, instruction for instruction, in TheNet (file `L2D.C`, `#ifndef PORTABLE`).

10 Two trivial routines, `ccphig` and `ccplow`, were added in TheNet to implement the `HIGH` and `LOW` commands. Each has 15 lines and comprises one if, three procedure calls, and a switch with two cases.

11 There are minor differences in other assembly language files related to NordLink's use of a later version of the C compiler (the Q/C compiler supports in-line assembly language). For example, the newer version of the compiler can save one byte when clearing a double register. In some cases, TheNet used a variation on the subroutine entry macro.

12 TheNet uses a `#define` statement in its primary include file, `ALL.H`, to define a preprocessor variable

FIRMWARE. When this variable is defined, the source code is conditionally compiled into TheNet (the network node controller), and when not defined, the code compiles into TheFirmware, a replacement for the user firmware for the TNC-2. The NET/ROM source is similarly structured with a preprocessor variable, and conditionally compiles the WA8DED AX.25 user firmware for the TNC-2. That firmware is available on many BBS, and is the foundation on which NET/ROM was built by the author.

13 TheNet does not contain the code to support the PK-87 TNC. NET/ROM's support for the PK-87 is conditionally compiled when a preprocessor variable called PK87N is defined.

14 NET/ROM, in my opinion, is concise and easier to follow (notwithstanding TheNet's extensive documentation in German).

Object File Comparison:

I have not personally evaluated the hex files of the original and the NordLink versions. Members of NordLink on at least two occasions have publicly suggested independent comparison of the binary files. However, they never recommended comparison at the source code level. Many well-meaning people in the U.S. have performed their own evaluations of the programs' differences based on the only materials available to them, the hex files. Their conclusions have ranged from "maybe 20-30 percent identical" to "definitely a copy." However, any judgment of the similarities of NET/ROM and TheNet from the comparison of hex files is fallacious because of the following:

- o A single difference in the relative placement of any global, local, or static data item (simple item, table, structure, etc) will render slightly different byte or word addresses. Since addresses comprise a major

portion of the object code, the hex address of the item will be different throughout the module.

- o A minor addition or removal of code (full duplex, HIGH and LOW commands, callsign encryption) will show as blocks of dissimilar code including addresses of function entry points, followed by major discrepancies.

- o Even a minor reordering of object modules in the linking step will render major differences in the hex file. Sophisticated pattern matching programs may be able to discover this reordering, however, jump addresses and procedure entry points beyond the reordering point will change significantly.

There is no possibility that the source programs for NET/ROM were obtained by NordLink as they had never left the author's house until the electronic version was loaned to me for review. The only real determination of whether TheNet is an original work can only be done at the source program level.

Evaluation:

Based on a line-by-line comparison of the two products and 22 years of software experience, I am convinced that the only way that TheNet could be identical in the structure, calling sequences and variable definitions of NET/ROM would be to have disassembled/de-compiled the object code from NET/ROM. TheNet is not an original development but rather a replica of the thoughts, concepts, and the painstakingly developed design embodied in NET/ROM. According to NordLink, "disassembling NET/ROM and then rewriting it in C would be silly." However, since the source was not available, their only alternative was to do exactly that - disassemble the binary code from a NET/ROM 27C256 EPROM and construct a source program that would produce identical binary

code.

Disassembly and De-compilation Methodology:

Without doubt, the starting point of this effort began with the low memory and Q/C library routines, and the routing table structure and the layer protocol definitions described in the NET/ROM documentation. The hex files of WA8DED's user firmware available in the public domain no doubt provided a convenient set of low-level I/O routines.

Generating assembly language from object code is relatively simple; disassemblers for all machine codes have been around a long time. Converting assembly language to a higher order language like "C" requires much more forbearance.

The Q/C compiler traces its heritage to Ron Cain's Small-C from the 8080 CP/M world (Hendrix "A Small-C Compiler"). It is a non-optimizing compiler and, consequently, the structure of its generated object code for any C construct is predictable and consistent. With suitable automated tools, much manual intervention and an intimate knowledge of the compiler's code generator, any section of code suspected of originating from this C compiler can be reconstructed into a syntactically correct source program.

Any programmer who has delved into compiler-generated object code will recognize that variable names and function names do not exist at this stage, merely address references to data and subprograms. However, if meaningful names are assigned to those addresses, and suitable comments placed in the source code, the original meaning and intent of a function in terms of a network controller will iteratively become evident. I say iteratively because a source program, when compiled, can eventually be modified to generate a given object program. When all object modules are linked in the same

order as the thing you're copying, an identical executable module will result. Minor changes of data location, removing callsign encryption, adding full duplex or other minor features could confuse a (hex file) comparison program, leading one to erroneously conclude that the executable modules are very different.

Source Code Comparison:

My visual examination of each routine showed that source code from both programs was identical, statement by statement, with only variable, data, and structure names changing. However, the source code does not lend itself well to comparison by automatic means. Because the object code was analyzed and equivalent source code was reconstructed from it, virtually no procedure names or variable names are the same. To perform even a cursory quantitative evaluation, one would have to remove all comments and white space from both versions, transliterate variable and procedure names into common, but arbitrary, names and convert both sources to either upper or lower case before a programmatic comparison could be attempted.

Additional problems thwarting an automatic comparison was TheNet's:

- o use of typedef, for example, 'typedef int VOID' and 'typedef unsigned BOOLEAN', which created synonyms for common data types

- o use of #define to create new language constructs, for example, '#define LOOP for(;;)' for an infinite loop

- o use of numeric constants in the source whose meaning was not necessarily understood. On the other hand, both programs made considerable use of #define to give (different) names to important and

frequently used constants

- o source coding variations when using a different set of data structures (note: code generated to access the data was the same as NET/ROM's accessing its own data structures!).

However, even after the automatic comparison, a manual examination would still be necessary to resolve differences such as:

```

        if (a != 0) {...}
and     if (!a) {...}

```

as being equivalent and identical, and to recognize that code segments such as

```

        a = b;
        for (i=0; i<max;
++i, ++a) {...}

```

```

and     for (a=b, i=0; i<max;
++i, ++a) {...}

```

```

or      for (xyz=foo, w=0;
w<limit; ++w, ++xyz) {...}

```

are entirely equivalent and would compile to identical object code.

As a better example of this comparison difficulty, consider NET/ROM's layer7 routine, validc, and TheNet's routine, fvalca, which validate a callsign:

```

validc(call, valflg)
char *call;
unsigned int valflg;
{
    return(*call == ' ' ?
FALSE :
        (valflg == FALSE ?
TRUE : valcsc(call)));
}

```

```

fvalca(pflag, call)
char *call;
BOOLEAN pflag;
{
    if (*call == ' ') re-
turn(0);
    if (!pflag) return (1);
    return (valcal(call));
}

```

These equivalent routines return 1 if the callsign is valid; 0 or -1 if not valid. Although they look quite different to the untrained eye, the curious programmer is invited to pass these examples through his favorite C compiler (I used Borland's Turbo C) and generate the intermediate assembly language; you don't necessarily need to target to a Z-80 or to a non-stack machine. These listings are identical if the formal argument parameters in fvalca are reversed, TRUE and FALSE are defined as 1 and 0 respectively (as they are in NET/ROM and TheNet with #define statements), and typedef'ing BOOLEAN as unsigned (as done in TheNet). Other less trivial examples I have run through my compiler show the same consistent comparisons at the assembly language level.

One of the more complicated routines extracted from both versions was the level 4 receive function l4rx (TheNet file TNL4.C) and l4rcve (NET/ROM file LAYER4.C). This particular pair of procedures was selected because it was representative of an extensive use of C structures and pointers. I was careful to insert (#include) the same files used in the parent source file and to reverse the arguments in TheNet's function calls before compiling. There were five minor differences in the 631-line assembly language files produced. The object file length for NET/ROM was 2599; TheNet's was 2577.

This slight difference can be attributed to my use of a compiler that is targeted to the 8086 family, stack-oriented processors unlike the Z-80; it merely is the only C compiler I have. As mentioned previously, Q/C is not an optimizing compiler and it produces code that is not stack-oriented. Optimization is standard for my compiler and cannot be disabled. Minor source coding variations can account for the order and manner in which addresses are cal-

culated.

Conclusion:

It is my conclusion, and I believe would be the conclusion of any rational reviewer, that TheNet is not an original development but rather a direct copy of NET/ROM. This exercise has left no question in my mind about the method that NordLink used to make their "original" design fully compatible with NET/ROM. Rather than start with the description of the layer protocols and the routing table, and then independently design and build a compatible product (as the author hoped somebody would), they disassembled Software 2000's product and reused the design in its entirety, procedure by procedure, and steadfastly proclaimed original work. According to NordLink, "it is truly a new and innovative program with many new features". I have seen no evidence of originality, innovation, significant enhancements or functional changes.

Thomas M. Allen, WA6IGY
CIS [72537,1143]

OIMO.EXE : Mailer for KA9Q TCP/IP software package

by Shigekei Matsushima, JK1RJQ
1-4-25 Sakurazutsumi
Musashino Tokyo 180 Japan

Internet:
shigekei%is.titech.junet@relay.cs.net

CompuServe: via JF1LZQ
[74600,276]

"OIMO" is the mailer developed by Shigekei Matsushima, JK1RJQ and Dai Yokota, JK1LOT for the user of KA9Q TCP/IP software package. This mailer has some new features that the obsolete version of BM.EXE does not have. Now in Japan, most of the TCP/IP'ers use this software as their mailer. "OIMO" has the

following features.

1. KANJI code supported
2. alias addressing
3. "autosave", save mails in the folders automatically
4. being able to invoke editor
5. source codes available
6. free of charge in case of non-commercial use

Let me explain some of them.

1. Many kanji codes are being used in Japan and they are not compatible. However most of the PC's running MS-DOS are using Microsoft Kanji code, so called Shift JIS, as their internal code. So most of the communication between PC's are made by Shift JIS code. This Shift JIS code doesn't appropriate for TCP/IP, because it employs full 8 bit. Then Japanese TCP/IP'ers decided to use JIS code in place of Shift JIS code, if they communicate through TCP/IP, following the example of the academic network JUNET.

Accordingly, the mailer has to have the filter which change the code automatically from Shift JIS to JIS kanji code. Our "OIMO" perfectly supports two kinds of kanji codes, JIS & Shift JIS Kanji code. User can select the output code by defining it in the start up file, called "oimo.rc". The only thing that user has to do is write mails on the PC's running MS-DOS. Even if user uses Shift JIS to write mails, the mails would be automatically changed into JIS Kanji code by default. Of course, if Shift JIS was designated as the output code in the "oimo.rc", mails would be left intact.

After the mails which were written in JIS Kanji code are received, when you read them, the mails are changed into Shift JIS Kanji code.

2. User can use the alias address in place of the long address. They are defined in the "oimo.rc". The format is as follows.

```
alias [alias address] [true address]
```

e.g.

```
alias shige  
shige@jklrjq.ampr.jp@jflqkj.ampr  
alias dai  
dai@jkl1lot.ampr
```

You can define the alias address using other alias address. For example, if aliases are defined as follows and the mail is sent to "oimo",

```
alias shige  
shige@jklrjq.ampr  
alias dai  
dai@jkl1lot.ampr  
alias oimo dai shige
```

the mail would be sent to shige@jklrjq.ampr and dai@jkl1lot.ampr. Don't care about the order of defining aliases. If you define the alias recursively, the recursiveness would be checked not to get into an infinite loop.

3. If user designate in the "oimo.rc" to save mails which includes some strings in their header, the mail would be classified and stored to the appropriate folder directory automatically with numbered filename. The designation format is as follows;

```
autosave [field] [string]  
[path] [?]
```

"field" is the one of the field in the header which "string" is placed. "path" is the full or relative path to the folder directory. If the relative path are used, the true folder is "FOLDER\path". FOLDER is defined in the "oimo.rc" or set as environmental variable. Last "?" is the option. This is the switch which decides whether "OIMO" asks the user or not, when mails are stored. For example, if user wrote the following designation in the "oimo.rc",

```
autosave from  
shige@jklrjq.ampr jklrjq  
autosave date Feb feb_mail  
autosave subject ka9q /
```

software/tcpip/ka9q

the mails from shige@jk1rjq.ampr would be stored into the folder "jk1rjq", the mails written in February would be stored into the folder "feb_mail", and the mails which concerns ka9q would be stored into the folder "\software\tcpip\ka9q". '/' and '\' are acceptable as the directory separator in the "oimo.rc".

The tools for managing mails in the folders are under programming.

4. User can use favorite editor while writing mails. If user writes a mail in reply, user can quote the mail which is replied with '>' at the top of the line in the editor.

Some of the messages that mail outouts and the manual documents are in Japanese now (February 4, 1989). However English version will be soon available. You can get a copy of "OIMO" from "oimo club". Please contact,

oimo club, c/o PRUG office
2-1-57, Shimohanazawa,
Yonezawa
Yamagata 992 Japan

If you want more technical information, write mails to,

technical dept., oimo club
c/o PRUG Tokyo branch office,
P. O. BOX 66, Tamagawa,
Setagaya
Tokyo 158 Japan

or

shigeki@is.titech.junet@relay.cs.net

through Internet.

Sorry for the Delay!

Your editor apologizes for the delay in getting this issue to you. After waiting until after Dayton to start this issue, time got away from us and we took too long to get caught back up. We're sorry. TAPR has begun searching for a new editor (p. 27) and I want to say thanks to all of the contributors who made this editors job as easy as possible!

Announcing a new release of the KA9Q Internet Package, revision 890421.1

This release constitutes an attempt to merge the best efforts of everyone how has been working on the KA9Q package since the last official release which was dated 871225.0. For those who've been tracking the beta releases, this package was built from 871225.33alpha.W9NK.4.1, with many additions. All users are encouraged to upgrade to this release as soon as possible.

Developers should be aware that this package likely represents the last official release of the KA9Q package until Phil finishes his internal rewrite to include a multi-tasking kernel, now known as the "NOS" version of NET. All development effort for new applications should be directed towards NOS.

Revision 890421.0 was distributed in a limited fashion on PC floppies at the Dayton Hamvention. For PC users, there is no appreciable difference between .0 and .1, other than the addition of modem dialing for slip, though the documentation has been somewhat improved.

The things that have changed since the 871225.0 release are too many to remember, much less mention, but here are a few highlights:

- addition of official support for the Atari ST, NEC PC-98XX, HP Portable Plus, and various System V Unix systems in addition to the PC and its clones.
- support for the FTP, Inc., packet driver specification on the PC
- support for IP transport over NET/ROM networks, and some NET/ROM user level functionality
- prompting for username and password in the FTP client
- a Finger application, similar to

Berkeley Finger

- an AX.25 mailbox
- addition of support for the W2XO PBBS when running under System V Unix, using SysV IPC between NET and XOBBS
- A complete rewrite (still rough, unfortunately) of the documentation
- conversion to the Borland Turbo-C 2.0 Professional Package for compiler/assembler/linker on the PC. This was done in response to heavy demand from the user base, and sets the stage for exclusive use of TC 2.0 in NOS. The package "almost compiles" under Aztec C 4.10d, and can probably be made to work... I just don't have time.
- addition of support for the MIT sifp serial line framing protocol
- modem dialing for slip and stfp

Contributions to this release came from *many* folks around the world, again too many for me to remember or mention. Special thanks are in order though for Bob Hoffman N3CVL who made this release possible by sorting through the muck and providing me with sources to the W9NK.4 version with SysV Unix merged in, and to Ron Henderson WA7TAS who made the Turbo-C 2.0 support work, added the HP Portable Plus support, and hopped in to do some dirty piece of work every time I was ready to give up in disgust...

HOW TO GET THE BITS:

Via FTP on the Internet:

The machine col.hp.com contains a copy of the distribution in the directory ~ftp/ka9q. Access is quite reasonable from other sites on the HP Internet, but "very" slow for folks outside HP. This site is recommended "only" for HP employees.

The bits may be found on tomcat.gsfc.nasa.gov in a directory somewhere under anonymous ftp called net-8904. This is a good place to grab the bits from right now.

The bits will make it to wsmr-simtel20.army.mil shortly. This is going to be the most stable Internet access point, I believe.

The latest alpha/beta release bits are frequently available from the machine louie.udel.edu, in the directory ~ftp/pub/ka9q, but users are warned that the code on louie is "guaranteed" to be broken in one way or another, so unless you're working on porting to a new target system or similar, "stay away" from louie.

Via UUCP or Phone BBS Download:

I no longer operate a telephone BBS system, nor do I support uucp from 'winfree' for grabbing the bits... my apologies for the confusion this has no doubt created.

Howard Leadmon, WB3FFV, has the bits available on his BBS, which also supports UUCP.

System Name: WB3FFV
Login: bbs
Number: (301)-335-0858 — 1200 & 2400 (Non-MNP)
Number: (301)-335-1955 — 2400 (MNP), 9600 & 19200 (PEP)
Data Settings: 8 Bits, NO Parity, 1 Stop Bit
Times: 24hrs/365days (except for routine maintenance)

Other folks also have BBS systems, if there's some other machine that you frequent for packet radio related software, check there first, and look for some indication of the version number.

Via Mail:

The Tucson Amateur Packet Radio association (TAPR), is distributing copies of this release

on IBM-compatible 360k 5.25" floppies. They also can provide KISS firmware for the TNC-1 and TNC-2, and clones.

TAPR can be reached at:

TAPR
PO Box 12925
Tucson, AZ 85732
USA
+1 602 323 1710

TAPR continues to be a leader in packet radio research and development, working with AMSAT on the microsat packet satellite project and a joint DSP project. The 'Packet Status Register' newsletter is well worth the membership fee. TAPR supports us, please support TAPR!

HOW TO REACH ME:

In the past, I included my mailing address and telephone number in these release notes. While the list of return addresses and folks who have contacted me is fantastic and astounding, I find that the amount of time required to deal with phone calls and paper mail has gotten a little out of hand. Therefore, I must request that questions about this release be sent by electronic mail, which is easier to cope with on a time-available basis. I *do* answer my mail when a working return address is provided!

Internet: bdale@col.hp.com
UUCP: winfree!bdale
Comuserve: 76430,3323
Packet: N3EUA @ KAOWIE

73 - Bdale Garbee, N3EUA

Renew Your Membership!

TAPR doesn't send out constant reminders when your membership has expired. Our only way of communicating is the date on the address label for this issue. Please check it and renew if required. Your membership is very important.

PK232 and MicroSat - Attaching a PSK Modem

The AEA PK-232, an excellent multi-mode data controller, is very popular with the satellite/packet community. At Dayton this year, a number of folks asked me how to interface the PSK modem to the PK-232. AEA made provision for attaching an external modem; unfortunately, the modem disconnect is not suitable for the TAPR PSK modem.

The PSK modem requires the following signals from the PK232 for proper operation:

- (1) 32X clock. This is generated by the 8530 and is available at U7 pin 14. It is not provided at the modem disconnect on the PK232.
- (2) Transmit data. This signal is generated by the 8530 and is available at U7 pin 15. It is also available at the modem disconnect on the PK232.
- (3) Ground or common. This signal is available at the modem disconnect on the PK232.

The PSK modem generates the following signals which must be applied to the PK232 in lieu of the signals generated within the PK232.

- (1) Receive data. This signal goes to the 8530 at U7 pin 13 (and 18). This signal may be introduced at the PK232 modem disconnect, and may be isolated by moving a jumper on the PK232 main PC board.
- (2) DCD. This signal goes to the 8530 at U7 pin 19. It may be introduced at the modem disconnect of the PK232 and isolated by moving a jumper within the PK232.

The PK232 modem disconnect also provides a PTT signal which is not needed by the PSK modem.

There are two primary approaches to connecting the PSK modem to the

PK232. The first is to modify the existing modem disconnect and tailor it to the PSK modem requirements. The second is to add a "standard" TAPR modem disconnect to the PK232.

MODIFY EXISTING MODEM DISCONNECT

The PK232 has a rear-panel 5-pin modem disconnect. It may be used with the TAPR PSK modem as follows:

- (a) Connect PK232 pin 1 (Rx Data) to the green wire of the PSK 8-pin DIN cable.
- (b) Connect PK232 pin 2 (Tx Data) to the blue wire of the PSK 8-pin DIN cable.
- (c) Connect PK232 pin 3 (DCD) to the yellow wire of the PSK 8-pin DIN cable.
- (d) Connect PK232 pin 4 (GND) to the red, black and shield wires of the PSK 8-pin DIN cable.
- (e) Connect PK232 pin 5 (PTT will become X32 clock) to the brown wire of the PSK 8-pin DIN cable.

Modify the PK232 circuit board by opening the PK232 cabinet (be careful you don't break the wiring to the battery holder in the top cover of the PK232 cabinet) and:

- (f) Carefully (l) cut the trace on the top of the board behind the modem disconnect that goes from J8 pin 5 towards the front of the PK232.
- (g) Solder a small-gauge wire from the feedthrough still connected to J8 pin 5.
- (h) Solder the other end of this wire to U8 (74LS393) pin 13.
- (i) Move the three jumpers JP4, JP5 and JP6 so they connect the center and rearmost pins together.

You are now ready to operate with

the PSK modem.

To return to normal operation, you must swap JP4 - JP6 back to shunting the center and forwardmost pins.

If you want to use the PSK modem's front panel switch to toggle between the PK232 internal modem and the external PSK modem, additional modification will be required.

- (j) Place the jumpers JP4, JP5 and JP6 so they connect the center and rearmost pins (same as step (l) above).
- (k) Solder the orange wire from the PSK modem 8-pin DIN cable to the free pin of JP4.
- (l) Solder the white wire from the PSK modem 8-pin DIN cable to the free pin of JP6.

The PK232 is now interfaced to the PSK modem and the PSK modem can select between the PK232 internal modem or the PSK modem.

ADD A STANDARD TAPR MODEM DISCONNECT

After doing these mods, the PK232 was starting to look like a repository for rainbow ribbon cable! In addition to the PSK modem which was now "permanently" attached to the PK232, the State Machine DCD Upgrade which I had hastily installed prior to Dayton was sitting on a RAM chip with octopus-like tentacles reaching all over the PK232 PC board!

Clearly, something had to be done.

The solution was to devise a PC board which plugs into the 8530 (U7) socket on the PK232 and adds a standard TAPR TNC modem disconnect header. This I did, also adding a connector for the State Machine upgrade.

The PK232 Standard Modem Disconnect is described elsewhere in

this issue of PSR.

Now I have the PSK modem set up for a TNC 1 configuration, and can simply unplug it when I want to move the PK232. In the future, I can add external modems to the PK232 knowing that they will plug right in if they are designed for a standard TAPR TNC! And, the State Machine DCD PC board and the PK232 Modem Disconnect PC boards look like they were meant to be installed in the PK232, not tossed in by a mad experimenter...

PK232 Standard Modem Disconnect Upgrade

In order to facilitate adding an external modem, as well as clearly installing the State Machine DCD Upgrade, TAPR has developed a PK232 Modem Disconnect Upgrade.

Construction involves soldering five parts onto the upgrade PC board. Installation consists of removing the 8530 from its socket in the PK232, plugging the 8530 into the Upgrade, then plugging the upgrade into the now empty 8530 socket!

The Upgrade provides a standard 20-pin TAPR modem disconnect header with the signals and disconnects needed to interface an external modem. It has been tested with the TAPR PSK modem kit and works perfectly.

In addition to providing the disconnect, this kit also provides an 8-pin site for directly attaching the DCD State Machine upgrade kit. A series resistor is mounted on the PC board for driving a front panel LED.

After installing the Modem Disconnect, the PK232 is ready for attaching the TAPR PSK modem for satellite and other weak signal work. With the State Machine DCD Upgrade also installed, the PK232 is ready to handle terrestrial packet chores more optimally (see elsewhere in this PSR for a synopsis on

DCD performance).

The Modem disconnect provides the following signals:

Pin 1 Carrier Detect Input

This pin tells the 8530 radio port that a valid data carrier has been detected. It should be pulled high when no carrier is detected and low when a carrier is present. This line must be implemented. It is jumpered to pin 2 when the PK232 internal modem is used.

Pin 2 Carrier Detect Output

This pin is an output from the the PK232 on-board modem and satisfies the requirements outlined for pin 1 above. It is jumpered to pin 1 when the PK232 internal modem is used.

Pin 3 (not used)

Pin 4 (not used)

Pin 5 PTT Output

This signal is used for transmitter activation. The 8530 will pull this output low when the PK232 wants to transmit; otherwise it will remain high. This pin is jumpered to pin 6 when the PK232 internal modem is used.

Pin 6 Transmitter Key Input

This signal is an input to the PK232 internal modem. It activates the PTT pin of the radio connector via the watch-dog timer. It should be left high and pulled low only when transmission is desired. This pin is jumpered to pin 5 when the PK232 internal modem is used.

Pin 7 (not used)

Pin 8 (not used)

Pin 9 (not used)

Pin 10 (not used)

Pin 11 Transmitter Clock (32x) Input/Output

This pin is tied to the 8530 32x clock output which generates a clock signal at 32 times the desired radio port data rate, e.g., 4800 Hz for 300 baud. The current PK232 software sets the 8530 to output this clock signal - software modifications would be needed to allow this to be used as a clock input.

Pin 12 (not used)

Pin 13 Receive Clock Input

This pin is tied to the 8530 receive clock input pin. It expects a clock at the desired data rate (1200 Hz for 1200 baud), of the proper phase relationship to the received data. This pin is normally jumpered to pin 14 when the PK232 internal modem is used.

Pin 14 Receive Clock Output

This pin is the received data clock signal. It is produced from a divide-by-32 chip (74LS393) in the PK232 digital section. This pin is jumpered to pin 13 when the PK232 internal modem is used.

Pin 15 PK232 Ground Reference

This pin ties to the PK232 digital ground.

Pin 16 (not used)

Pin 17 Receive Data Input

This pin is the received data input to the 8530. In the PK232, this is applied to the normal Rx Data input pin as well as the CTSA input for bit-banging the non-packet receive modes. This pin is jumpered to pin 18 when the PK232 internal modem is used.

Pin 18 Receive Data Output

This pin provides receive data

from the PK232 internal modem. This pin is jumpered to pin 17 when the PK232 internal modem is used.

Pin 19 Transmit Data Output

This line is the NRZI data output from the 8530. This pin is jumpered to pin 20 when the PK232 internal modem is used.

Pin 20 Transmit Data Input

This input line accepts data to be transmitted by PK232 internal modem. This pin is jumpered to pin 19 when the PK232 internal modem is used.

WARNING!

Note that ALL modem disconnect signals are at TTL or CMOS levels, NOT RS232! DO NOT CONNECT ANYTHING TO THE MODEM DISCONNECT WHICH IS OTHER THAN TTL COMPATIBLE OR SERIOUS DAMAGE MAY RESULT TO YOUR PK232!

If you elect to use an off-board modem, be sure to properly shield the interconnecting cables for RFI protection. The TAPR PSK modem interconnect cable supplied with the PSK modem kit is a shielded cable.

DCD MODIFICATION KITS NOW AVAILABLE FROM TAPR

by Lyle Johnson, WA7GXD

(Also see "Hardware Available from TAPR", this issue)

IMPROVED PERFORMANCE IN YOUR LOCAL AREA (AND HF, TOO!)

THE "DCD MODS"

BACKGROUND

Proper operation of Data Carrier Detect (DCD) is imperative for effi-

cient sharing of a packet channel. Many TNC's don't provide optimum DCD operation, and the current version (2.0) of AX.25 Level 2 protocol compounds the problem.

However, an inexpensive solution is now available to combat the former case - and progress is being made in the latter case with the proposed changes to AX25 Level 2 Version 2.1.

THE PROBLEM

The Tucson LAN operates via a mountaintop repeater dedicated for packet use. With a radius of coverage approaching 200 miles, it is essential that all stations be able to properly detect use of the channel by other stations and defer their transmissions until the channel is clear.

Over time, it has become apparent that most modems are lacking in proper DCD operation. Some are much worse than others. Some are OK, but allow improper operator adjustment without letting the operator know the "Threshold" adjustment is incorrect. (TNC2 code release 1.1.6 alerts the operator by not passing along packets that are received if DCD was not activated. This encourages the operator to properly set any DCD threshold control that may be on his TNC).

Eric Gustafson, N7CL, has done extensive investigation into this problem and presented his findings at the 7th ARRL Computer Networking Conference last fall. Most of the same information has also been presented in the recent PSRs.

SOLUTION

If the DCD decision could be made on the basis of "information coherence" rather than "is there some sort of signal or noise present?", LAN operation will improve. This premise has been demonstrated in a number of locations where modifications to TNCs have been made.

Inexpensive kits are now available from TAPR to make it a trivial matter to upgrade most TNCs to improved DCD operation. Check the article in this issue which lists the hardware available from TAPR for price and ordering info.

THESE MODS ARE EXTREMELY USEFUL FOR BOTH VHF AND HF OPERATION

The 2211 DCD Upgrade Kit
\$11.00 from TAPR

For TNCs using the XR2211 demodulator such as:

AEA PKT-1
AEA PKT-80
DRSI HF*MODEM
GLB PK-1
GLB TNC-2A
Heath HD-4040
MFJ 1270
MFJ 1274
PacComm TNC-200
TAPR Beta Board
TNC 1
TNC 2

The PC board is tiny, less than 2 inches on a side and shaped to fit into a TNC 1 or TNC 2. After building it, you simply unplug the XR2211 chip from its socket, insert it into the socket on the upgrade board, then plug the upgrade board into the IC socket vacated by the XR2211 chip on the TNC.

If you are into HF operation, provision is made to connect a "Threshold" control onto the demodulator.

The result will be fast-attack, slow decay DCD with a hang time to compensate for temporary fades due to multipath. When all stations sharing a channel have proper DCD action, data flows more efficiently.

The State Machine DCD Upgrade Kit
\$17.50 from TAPR

For TNCs with other modems such

as:

AEA PK-87
AEA PK-88
AEA PK-232
AIWA APX-25
AIWA APX-25M
DRSI PC*PA Type 1
DRSI PC*PA Type 2
Heath Pocket Packet
KAM
KPC-1
KPC-2
KPC-2400
PacComm Tiny-2
PacComm TNC-220
TASCO TNC-20
TASCO TNC-20H

The upgrade adaptor for these TNCs adds an EPROM-based State Machine to derive DCD based on lockup of a digital phase-lock loop. It is a PC board less than 2 inches square, and mounts easily inside the cabinet of most any TNC (NOT the Heath Pocket Packet/TASCO TNC-u21).

This upgrade will **DRAMATICALLY** improve DCD operation, even allowing you to run your radio unscelched which reduces other stations' TXDelay requirements, further improving throughput on the channel.

Errata Sheet - DCD State Machine Upgrade

There are some errors in the DCD STATE MACHINE DOCUMENTATION.

1) The 7910 interfacing section incorrectly states that Receive Data is available at pin 24. The correct pin is 26.

2) Suggested Kantronics KPC-2400 interfacing is:

U3 - (7910 chip)

Receive Data - Pin 26 (GRAY).
Carrier Detect - Pin 25 (VIOLET).

U6 - (4024 divider)

+5V - Pin 14 (BROWN).
x32 Clock - Pin 5 (BLUE).
GND - Pin 7 (RED).

U19 - (HD63B03XP microprocessor chip)

Lift pin 19 and attach DCD output (GREEN).

Place the DCD jumper at JMP pins 1 and 2.

3) AEA PK-232

There are a number of errors in this interfacing section.

Remove push-on shunt at JP6, not JP5 as stated.

Receive Data is available from U15 pin 6 (GRAY).

X32 clock is available from U8 pin 13 (BLUE), not U18 pin 13 as stated.

+5 is available from U8 pin 14 (BROWN).

Ground is available from U17 pin 7 (RED).

DCD is available from the 5.1K resistor end nearer the C57 silkscreen legend (VIOLET).

DCD out may be attached at U7 pin 19 (GREEN).

NOTE: The DCD modification may affect Morse reception at certain speeds. It seems to not affect AMTOR operation or BAUDOT. It is recommended that an additional LED be mounted on the PK-232 front panel per the instructions given. Proper tuning of signals in all modes will be easier if this is done. It is important that the new LED as well as the old DCD LED be illuminated during MORSE and BAUDOT operation for proper decoding by the PK232.

TAPR NOW PRODUCING TNC-1 UPGRADE KITS

NOTE: The first production run of the TNC 1 upgrade kits were sold out at Dayton. The second production run is expected to be available in mid-June. Price and ordering information appears in the article "Hardware Available at TAPR" in this issue.

Overview

The TNC 1 Upgrade adds an enhanced TNC2 to the TNC1 chassis. When the upgrade is completed, the TNC will have all the capability of the TNC 1 coupled with all the capability of the TNC 2.

Looking at it from the TNC 2 perspective, the upgraded TNC 1 provides all TNC 2 features plus the following new ones:

- (a) Software selectable serial port (ABAUD) and radio port (HBAUD).
- (b) Two sets of default parameters in battery-backed RAM (optional).
- (c) Two sets of EPROM-based software (optional).
- (d) Complete TNC2 firmware capability (NET/ROM, for example). This also "ensures" availability of firmware for the upcoming AX.25V2.1, etc.
- (e) Two modem disconnect headers (one for the TNC 1, one for the TNC 2).
- (f) Front panel RESET switch.
- (g) A TNC 1I.

Upgrade Description

The upgrade is a kit that can be built in an evening or two, depending on the builder's skill, experience and manual dexterity (had to get at least one four-syllable word in here).

After construction, the unit installs in the TNC 1 by removing the UART chip from the TNC 1 (6551, U14) and the push-on jumpers at the modem disconnect header (J5). The upgrade PC board plugs into U14's

socket and J5. U14 is then installed on the upgrade board and the push-on jumpers installed in the new TNC 1 modem disconnect (T1).

A four wire harness is installed on the TNC 1 and plugged into the upgrade board.

For those that own early TAPR "Beta" TNCs, the upgrade will work with these too! Installation is a little trickier, but not overly difficult.

Baud rates are set by selecting the TNC 1 and setting the rates by issuing the ABAUD and HBAUD commands (or <ESC>>B and <ESC>H if running the WA8DED firmware in the TNC 1. The TNC 1 baud rate generators then control the TNC 2 baud rates.

Like the latest TNC 2s, the upgrade accepts 27256 EPROMS for firmware, and uses a 32k byte static RAM chip for all RAM functions.

A second RAM chip (8k bytes) may be installed to allow selection of two sets of default parameters (two call signs, or HF and VHF settings, etc.). Since the TNC 2 uses the lower part of memory for parameter storage, a smaller (cheaper!) RAM is used as the second RAM. The remaining 24k bytes of the 32k byte RAM space is then "borrowed" from the primary RAM chip.

Of course, you may elect to install a 32k byte second RAM chip, in which case the upgrade unit will use the entire 32k bytes of it. The second RAM chip, regardless of size, is an option.

A second 27256 EPROM may likewise be installed to allow two sets of firmware to run in the TNC 2.

A local reset of the upgrade processor automatically occurs when you switch between banks (the TNC 1 is not reset in this case—you must manually press the RESET switch on the front panel).

Errata Sheet - TNC 1 Upgrade

(This is an updated Errata Sheet for the TNC 1 Upgrade. Item 10 is new and applies to those of you who are upgrading a Heath HD-4040.)

Page 4

- 1) Be sure to install the 28-pin IC socket called for at U14, not P11
- 2) U8 is called out as 16-pin; it should be 14-pin.
- 3) U9 is called out as 14-pin; it should be 16-pin.

Page 5

- 4) Resistor leads are on 1/2" centers. A lead forming jig or an ALJ-1000 (for you early TNC 2 kit builders!) may be useful for neat construction.
- 5) C1 and/or C2 may be 22 pF (marked 22 or 220).

Page 6

- 6) JMP1 through JMP5 are labeled JP1 through JP5 on the PC board.
- 7) When installing T1, be sure you do not install it at J11

Page 9

- 8) Under instructions for Heath HD-4040, C14 (on the TNC PC board) must be a low-profile capacitor. If it is taller on the PC board than the pins or shroud of J5, it will need to be replaced with a low-profile part before proceeding with construction of the upgrade.

Page 14

- 9) Last step of PRELIMINARY TEST (bottom of the page). After removing the Upgrade PC board from the TNC, remove the temporary jumper between R1

and ground.

Page 17

- 10) Fourth to the last step. The trace to be cut is between U6 pin 39 and U21 pin 11.

Page 18

- 11) The second to the last step should read "All ICs OK."

Page 20

- 12) The TNC 2 with TAPR firmware will default to AWLEN 7, even parity.
- 13) The BANK switch must be in BANK 0 (TNC 1) for the upgrade to come out of reset.
- 14) If you wish to operate the upgrade with no external front panel switches or controls (omitting the 4-wire harness, for example), set the jumpers to: JP2 and JP4 @ 2-3; JP3 and JP5 @ 1-2 and install the 32k RAM at U5 (not U4).

8th Network Conference

(continued) from page 5

The RMPRA Packetfest will feature many of the prominent amateurs in attendance for the 8th Networking Conference and the Digital Committee meeting. The Sunday session will be of the tutorial/discussion/Q&A type of presentations.

Conference headquarters will be at the new Colorado Springs Marriott Hotel. Special conference rates are: Single person in room \$45.00; Extra person in room \$13.00. Reservations should be made by September 6, 1989 at which time the reserved block will be released. After this date there is no assurance that space or the special rates will be available.

If making reservations by phone, call 719 260-1800 (do not use the Marriott 800 number for these rates) and ask for Reservations. Be specific in identifying yourself as a member

of "ARRL Networking Conference". This is the only way that you can be assured of these rates when reserving by phone.

The Colorado Springs Marriott is a new hotel, having opened in March 1989. A spacious and very pleasantly appointed Conference Hospitality Suite will be available from Friday afternoon until Sunday noon.

Colorado Springs Municipal Airport is approximately 10 miles from the Marriott. It is served by six national carriers with over 100 flights daily from six major gateway cities: CHICAGO, ST. LOUIS, DALLAS/FT. WORTH, PHOENIX, SALT LAKE, DENVER. Airport transportation will be provided by the Marriott to and from the airport. Arrival and departure times may be coordinated with the hotel transportation staff.

For those arriving before 3 pm on Friday, October 6th a conducted tour of the Air Force Academy has been planned. The assembly point for this tour will be announced in a later bulletin and will be posted at the hotel registration desk. Private transportation will be used. If you need a ride let your needs be known when sending in your registration fee.

The registration fee for the conference is \$20.00. This fee includes the conference, one bound copy of the 8th Networking Conference proceedings, refreshments throughout the day, lunch at the AFA Officers Club and use of the Marriott hospitality room. There will be no charge for the conducted tour of the Air Force Academy. Extra copies of the conference papers will be available for \$12.00.

Upon receipt of your registration fee you will be mailed a pre-printed Marriott reservation form and other material of interest. Please indicate if you would like to be included in the Air Force Academy conducted tour. Send \$20.00, (make checks payable to Andy Freeborn) your

name, call, address and telephone number to:

Andy Frøeborn, N0CCZ
5222 Borrego Drive
Colorado Springs, CO 80918
Telephone (719) 598-8373

TAPR's packetRADIO

At the Dayton Hamvention in April, TAPR demonstrated two radios designed especially for packet. To explain the reasoning behind their design a brochure was made discussing the engineering criteria behind the prototypes. 3500 brochures were printed and before the end of the convention they were all gone. Since then many individuals and manufacturers have requested information on the project. Alas, the brochures are gone.

Stan Horzepa, WA1LOU, editor of Gateway, ARRL's biweekly packet newsletter, gave a full report on the project. Stans report was comprehensive and factual. In the absence of additional brochures to distribute we're reproducing the Gateway article here.

From GATEWAY:
The ARRL Packet-Radio Newsletter Vol. 5, No. 17 May 12, 1989

packetRADIO - THE MISSING LINK

Something has been missing from packet radio... a radio designed specifically for the average amateur. Why 9600 bit/s? Most VHF amateur packet-radio operations use 1200 baud modems with radios designed for voice use. With few frequencies available for packet-radio use, today's channels are becoming extremely crowded. Packet radio is almost unusable in some metropolitan areas during evening hours. Sending data faster allows more users to operate on a given channel. And 9600 baud can be easily encoded using FSK techniques and fit within a normal 2-

meter FM channel. But, a faster modem running with a voice radio is a compromise at best. Why a Special Radio for Packet Radio? The typical VHF packet-radio station uses an FM transceiver designed for voice use. There are four major drawbacks to using such radios.

1) Timing. Voice radios have receive-to-transmit and transmit-to-receive turnaround times of about 150 to 400 ms. This dramatically reduces the amount of data that can be sent and increases the chance that two or more stations will interfere with one another. At 9600 baud, a radio that switches in 1 ms can transfer files about 20% faster than one which switches in 200 ms. Similarly, a channel can accommodate four times as many users if the radios switch in 1 ms instead of 200 ms. At data rates faster than 9600 baud, the differences are even more dramatic.

2) Interfacing. The modem-to-radio interface depends on audio response, filters and audio levels intended for microphones and speakers. More often than not, this leads to incorrect deviation of the transmitted signal, noise and hum on the audio, and so forth. Splatter filters and deviation limiters distort frequency response and further reduce the performance of the packet-radio system. Higher speed operation (such as 9600 bit/s) involves surgery on the radio - there is no proper interface. The TAPR packetRADIO has built-in 1200 baud and 9600 baud modems. It plugs directly into a standard TAPR TNC modem disconnect. Its filters are optimized for data operation, not voice.

3) Complexity. The typical VHF radio manufactured today is competing in the voice market and includes many additional features which are simply not necessary in a data radio. These include telephone tone pads, scanners, digital readouts, squelch, voice synthesiz-

ers and miniaturization. In fact, these additional features often detract from the performance of the radio in data applications.

4) Price. The usual VHF FM transceiver sells for \$400 or more in today's market. A dedicated digital transceiver can be made to significantly outperform existing voice-grade radios for data use and at a substantially lower price. It makes good economic sense to free up a multi-featured radio for voice operation and use a simple, inexpensive data radio for digital applications. TAPR's packetRADIO has the following features that are designed for experimentation.

- Design is easily adaptable for higher frequencies.
- Each major subsection of the radio is on a separate printed circuit board for optimum performance. This results in the ability to upgrade to other frequency bands.
- The modems are modular. Higher speed operation to 56 kbaud and beyond is possible.
- The basic RF deck may be used for modem experimentation.
- May be used with a transverter for higher frequencies and higher baud rates.

The packetRADIO also provides the following features for optimum performance.

- PIN diode antenna switch.
- 25 watts RF output.
- 5 crystal-controlled channels.
- 1200 baud AFSK FM operation for communicating with existing stations.
- 9600 baud FSK operation for performance (optionally capable of 19.6-kbit/s operation).
- Operates in the 144-146 MHz band (optionally 220-225 MHz). Higher frequencies will be available as low cost components enter the market.
- Plugs into standard TAPR modem disconnect.
- Capable of full-duplex operation with optional second local oscillator board.

- Switches between transmit and receive modes in less than 1 ms.

The radio also has simplified front panel controls:

- Power: on/off.
- Channel: steps through five available channels.
- Speed: toggles between low-speed (1200 baud) and high-speed (9600 baud or optionally 19.6 kbaud) operation.
- DCD: allows adjusting receiver performance to match channel requirements.

Front panel LEDs are used to indicate:

- DCD: Data Carrier Detect.
- XMT: transmitter activated.
- CON: connected to another station.
- STA: frames have been sent but not yet acknowledged.
- PWR: power is applied and the unit is switched on.

The following two modems are standard equipment:

- 1200 baud AFSK FM modem with optimized DCD, preset for proper deviation, for communication with existing users.
- 9600 baud FSK modem, compatible with existing 9600 baud FSK packet-radio modems (TAPR, G3RUH, TexNet).

Bits in the Basement

by Bdale Garbee, N3EUA

Adapted from the
RMPRA>PACKET

A friend contacted me via electronic mail recently, asking if I'd be willing to divulge a few secrets about what I've been working on. Though it at first seemed silly, his request made me realize that there are only a couple occasions each year when most packeteers get a real data dump about what's happening on packet's leading edge. I'm going to try and change that, by writing a regular column for this newsletter about

what's happening on the "edge of the envelope" in packet.

At Dayton this year, I helped man the TAPR booth with Greg Jones, WD5IVD of the TexNet organization. At one point, Greg told me that while I was out walking the flea, a fellow wandered up to the TAPR booth, looked back and forth for a minute, and then asked Greg "Is this really the TAPR booth?". Greg said that indeed it was. The fellow looked back and forth for another few seconds, and then said "but... you're just ordinary people!" True story. I hope the point I'm trying to make is obvious. There isn't anything mystical about the pioneers in packet, they just *do* things instead of talking about them. All it takes is imagination, and a lot of determination.

I will be satisfied if I accomplish nothing more with this column than to startle you into realizing just how much POTENTIAL there is for technical advance in packet. If I can stir up your excitement for actually playing with some of the new technologies I'm going to discuss, then I'll be really excited!

There are basically three things I want to talk about this month. The brand-new release of TCP/IP software, a project I've been working on with N6GN for 10Ghz packet, and a rundown on some new packet facilities that I'm putting on the air here in "Bdale's Bit Basement".

It's been about 16 months since the last official release of the KA9Q TCP/IP package on Christmas day in 1987. As the official integrator and distributor, I take most of the blame for that... little things like buying a house and getting married got in the way... But all that has changed, with the release of version 890421.0, which was available from the TAPR booth at Dayton, is now available on floppy from TAPR by mail, and is becoming available from "all the usual places" electronically.

The biggest changes since the last official release are addition of an AX.25 mailbox, the ability to do IP over NET/ROM, username and password prompting in FTP, a 'Finger' server for finding out information about users on the air, a new 'mheard' command like that found in TNC's, optional multi-port digitizing for switch sites, and support for more machines. We now support PC clones, the Mac, the Atari ST, HP laptops, NEC PC-98's, and a variety of variants of Unix System 5. There is also a new PBBS program by W2XO included that runs only under Unix, and links into NET, providing a host of really neat new features... the BBS code is a bit rough around the edges, but we're working hard on it. Expect to hear more about the new release at the Packetfest. Rest assured that, as I mentioned in an article here a few months back, just because we've been quiet doesn't mean we haven't been working hard.

Glenn Elmore, N6GN, and I have been working since about the beginning of the year on techniques for running 1-10Mbits/sec (yes, that's 10 million bits per second, Ethernet speed if that means anything to you...) on 10Ghz using direct FSK and Gunnplexers. I showed a 1Mbit, 10Ghz link in the packet forum in Dayton, which from bits to RF and back (with 2-foot dish!) is under \$100 per end. The technique involves using the digital data on transmit to "pull" a 3-terminal LM317 voltage regulator chip driving the bias/tune pin of an NEC ND751AAM Gunnplexer module, designed for use in police radar guns. This results in FSK modulation, as the varying tune voltage changes the operating frequency. On the receive side, we use two MMIC gain stages set up to do band-pass filtering around 105Mhz, then mix down to 45Mhz with an off-the-shelf mixer and single-transistor local oscillator, then use a single chip Motorola 13055 FSK receiver/demodulator to recover the data. A

quad op-amp provides for carrier-to-noise and tuning meter functions. We're using some surplus IBM networking cards to drive the link at 1Mbit.

There should be a writeup in a major ham magazine by fall. People, this stuff is FUN to play with! Using Gunnplexers means that the microwave part of the problem is effectively taken care of, and all you have to worry about is reasonable-frequency RF. Hopefully we will eventually work out some kind of arrangement for distributing PC boards and/or kits, but for right now we're still too busy tinkering. We want to try slower data rates (say, something near DC, like 100kbps) for longer path lengths (we expect 35-50 miles line of sight with 2-foot dishes from 1-5Mbits/sec), higher data rates (to 10Mbits) for local operation, and full duplex since the Gunnplexers work that way already.

Fundamentally, the 10Ghz links are point-to-point, which will make them ideal for backbone links with something like the PS-186 packet switch replacing our current back-to-back TNC's, or for local links between individual users who want to play with high-data-rate requirement applications, like digital voice and digital video. If nothing else, the availability of a technology like this for cheap means we need to rethink how we plan our network topology and links.

So, what am I actually running on packet these days myself? Well, I recently put a surplus HP9000/550 unix system on the air. I have two ports, both 1200 baud, one running on the Colorado 450 backbone to do BBS forwarding, the other one sitting on 145.01 for local access and a hook into the NET/ROM backbone for long distance TCP/IP activity. (It's interesting living in an area where 145.01 *isn't* crowded...) I am, of course, running the KA9Q TCP/IP package. I am also playing

with the W2XO PBBS package, which links to the TCP package and uses the same pair of KISS TNC's. I can have multiple TCP/IP sessions, multiple AX.25 sessions, and all kinds of other stuff happening at once.

The really neat part, though, is that with the new release of the TCP/IP package, we can run IP over the NET/ROM network. The way this works is that I have my station also set to act as NET/ROM node '#COSIP'. The pound sign causes the address to not be visible to normal users, since it isn't useful to them. The trailing 'IP' designates this node as a TCP/IP packet switch. Others who are online or are coming online to act as TCP/IP gateways in various areas include '#LAMIP' in NM, run by Gary Bender WS5N, and '#DENIP' in Denver run by Fred Schneider, KOYUM. The idea is that users in the Springs can set up their TCP/IP routing to point all traffic to nodes outside the local area through my system, which will make a decision about where to send the packets, and then route them to other clusters of TCP activity over the NET/ROM network.

For example, say N0CCZ in Colorado Springs wanted to send a file to someone in Denver. He'd set up his routing with a command to send all non-local traffic to N3EUA. Then, all he'd have to do is type 'ftp koyum', and the packets would flow from his station to mine, be switched onto the NET/ROM backbone destined for #DENIP, where they would be picked up and switched back to a local TCP/IP frequency by KOYUM's machine, and from there hit the Denver station. That station would have set up his routing to send packets destined for non-local addresses to go via KOYUM, and the reverse process would occur for return packets. This allows a very efficient usage of the NET/ROM backbone compared to normal users, because the backbone

is used purely for computer to computer communications, with the only open circuits being between the TCP/IP switch sites. I have quite successfully exchanged a bunch of mail with WS5N in the Los Alamos area by this technique, even though the round trip time on the path is on the order of 3.5-6 MINUTES... the excellent stability of the TCP protocol under adverse path conditions makes possible the use of extremely poor paths for mail transfer.

Other neat facilities that I'm close to providing on my system are gateways between the TCP/IP SMTP mailer, the PBBS mail forwarding network, and the rest of the networked world. I am already forwarding all incoming SMTP mail and PBBS mail for myself into the Unix mailer. Soon, I'll have the rest of the bits in place, and we'll be able to forward PBBS mail for TCP/IP users direct to their machines, and all kinds of other neat stuff! In addition, I now have a copy of the complete FCC call sign database online, and am working on mechanisms for allowing access to the data. Expect to see more about this in the near future.

I get asked frequently what is happening with the 56kbit modems. The modems themselves work wonderfully. John Conner W0DFHG and I have a pair pretty much ready to go. The problems have been RF and bits. On the RF side, we bought a pair of Microwave Modules 28Mhz <-> 430Mhz transverters (expensive!). At Dayton this year, John and I bought a pair of KLM 6-element yagis, which we've just gotten on the air, and it looks like we might finally be able to work each other directly. On the PC side, we spent a long time trying to do the GRAPES-specified hacks to a pair of TNC-2 clones to make them do 56kbits half duplex on the modem side, then gave up in disgust. The Georgia boys now have the TNC's in question, and they haven't gotten them working either... my personal opinion is that they are running the TNC's on the hairy edge,

and we should just look for other options. Short term relief has arrived in the form of a pair of DRSI PCPA cards that John and I purchased at Dayton. These drop into a PC and have one 1200 baud channel, and one set up for an external modem. Phil Karn KA9Q has written a driver to make them go 56kbits by basically shutting the rest of the machine down (disabling interrupts) and pushing bits in and out as fast as he can. We hope to use this driver with the new cards to put the two units we have now on the air in Colorado Springs for some real testing. More on that when it happens.

Longer term, Mike Chepponis K3MC in Fremont, CA, is working on a new I/O card for PC clones that includes an onboard processor and memory, designed specifically to drive up to two high-speed and two low-speed channels, taking much of the load off the host processor. This will be the card of choice in my mind for 9600 baud through 1Mbit speed links. Expect it to be commercially available, perhaps also as a kit from TAPR, by the end of the year.

The overall cost to put a 56kbit station on the air now appears to be about \$600, which compares really poorly with things like the 10Ghz units at \$100 and end plus, maybe, \$150 for a digital card. I'm still excited about the 56kbit units since they operate on a band where omnidirectional antennas are still possible, but with their high cost, I think we ought to consider heavily the use of the TAPR 9600 baud technology for local channels, with microwave backbone links...

I welcome comments and suggestions for topics for this column. Whether I try to keep it up on a regular basis ultimately will depend on the interest you express. 73!

Using Round Table Packet Systems for Emergency Communications

by David Cheek — WA5MWD
(Reprinted from May '89 TPRS Report)

The Public Service season in amateur radio is upon us. I'd like to present some packet tools that provide good real time communications. This is far superior to the "forwarding BBS" type network that can work well for long term operations, or health and welfare situations.

Some tools available for these activities are:

The round table on RLI type mailboxes.
The ARES DATA on line database and message system.
A mysterious contender, name not known at this time.

Most of these have one thing in common, a roundtable type of operation. The value they add is, that it is possible to send a message to a single operator and bypass the "roundtable function". This may seem to defeat its purpose, but in fact it is ESSENTIAL TO PACKET if it is going to help in an emergency. This is because it allows a form of packet "NET" with a net control station.

Everything the net control station says may need to be heard by all members. Some things the participants say, only need to go to one place. This helps cut down on non-essential traffic and every packet operation is always limited by the channel time available. If we had unlimited bandwidth and speed, we would not have to worry so much about bypassing this roundtable function.

The bypass usually requires a special command. The mystery system

uses the syntax: /MSG Callsign text of the message. Just sending without this causes broadcast and lots of traffic on channel. This means that net procedures and net DISCIPLINE are a requirement! The ARES/Data database uses this syntax to do the same thing; tell callsign text of the message. Sending the command without the callsign causes the message to go to all connected stations. Net discipline is easier to maintain. In both systems, there are ways to make subsets of the roundtable. Each of these methods of subsetting has its problems, so I think neither has an advantage.

Roundtable systems have one other thing in common; the method of finding out who is on the "table". This is usually a command. In the mystery contender, the command is: /who. In the ARES/Data system, the command is: users.

The big difference between these two is that the ARES/Data system requires a PC Clone and a TNC with WA8DED host mode software. Since I don't have that software, I've just investigated it from the sysop console. I suspect that it is slow. This system has a DATA BASE, running on the PC, which is its real value. This is described in more detail in the ARRL 7th Computer Networking Conference Proceedings. Let me just say that it is best suited for shelter management, people tracking, and general resource management. It is copyable for non-profit purposes.

The mystery contender is software that replaces the normal firmware in a TNC2 clone. This allows it to be mounted in the only place a conference bridge can work, near the center of all stations. All conference bridges have a weak point. If they are far from the users (connected by digipeaters), then the data rate slows to a crawl. Some users may be disconnected from the bridge without anyone knowing about it. This happens if they retry out with the

bridge sending text to them. I am not sure where this item came from. Message me if you have an interest. I WILL NOT make copies of the EPROM because I don't have it. I've just looked at one as a user.

I have not described the Round Table option on the RLI BBS, as it does not have a "directed message" option. It only does the one to all roundtable function. It is still very valuable in places where you don't have time or other resources to set up a better system. If you want to know more about RLI roundtable, ask your local RLI operator, or check in and give the command; then EscapeH will give you help (you won't do this on 145.01, will you? I knew you wouldn't). Notice that all the RLI commands require the escape character to precede the letter. This may be a problem for some terminal software. The thing I like about the RLI table is that it can link the multiple ports of the BBS. If you have multiple VHF/UHF ports, this can extend the geographic range of the net. I don't think the two other types can do this.

The RLI system and the ARES/Data system have an explicit control operator. The functions of this control operator are limited in both systems, but at least the idea is there.

I hope that you will experiment with these systems, as they seem to offer much more potential than just a TNC and a computer. Remember that a GOOD Signal, great antenna, and proper operating procedures are essential to making any packet radio operation a success. Without them, you might as well leave your rubber duck at home.

Packet Products

[Editor's Note: PSR welcomes information from vendors of packet radio equipment to amateurs. Please send announcement information to the Editor, PSR, in care of the TAPR office address listed on the back cover of this issue. PacComm made a series of new product announcements at Dayton. This information, provided by PacComm, summarizes their new products.]

PacComm announced two new TNCs on display - the PC-320 and the TNC-320. PacComm calls these the "320 Series" because they share the same firmware and modem design. The PC-320 is a 3/4 size plug-in card for the PC or Tandy-1000, while the TNC-320 is a traditional TNC in a cabinet for use with any RS-232 computer or terminal. Both units support HF and VHF/UHF packet operation with a separate modem optimized for each type of signal. The HF modems use the EXAR 2206 and 2211 style modem with a 6 pole filter in the receive circuit. The TNC320 model also features a front panel threshold control to take best advantage of the 'hang' feature of the carrier detect circuit and the built-in LED tuning display. The VHF/UHF modem uses the same TI 3105 modem chip used on the TINY-2 and MICROPOWER-2 packet controllers.

A TSR (background) program provided with the PC-320 provides a simulated display of the conventional 'LEDs' found on a standard TNC and a ten-element tuning display for the HF port. The status display can be moved to any position on the screen or made invisible while still providing a connect alarm.

Operating prototypes of the PacComm Narrowband 96 Packet System were on display. The Integrated

Packet Radio (IPR) features a TINY-2 packet controller, 9600 baud modem and commercial duty crystal controlled RF deck built into one case. The unit displayed was very compact, about 6" wide, 3" high, and 7" deep. Operation is very simple - merely connect an antenna and computer cable and apply power! Also on display were the NB96 Digital Radio which is like the IPR but without the internal TNC. The model shown attaches to any TNC with a TAPR style modem disconnect header. Another model will attach directly to the AEA PK-232.

There were a number of different models of the 2" x 3" UMPAD (Ultraminiature Packet Assembler/Disassembler) on display. One of the UMPAD units was installed internally in a Toshiba 1000 laptop computer with a cable connecting directly to a handheld radio. This unit was constructed by Fred de Bros, KX1S, and drew a continuous line of interested spectators. Development work on a production model for the laptop is underway, but a delivery date has not been established yet, nor has a decision been made about which other laptop computers may be supported.

PSR Editor Wanted!

Our current editor, W3VS, has asked that a new editor be found before the next issue of PSR is published. Scott has edited the last several issues but is unable to continue due to other commitments. If you've got an interest in packet and are willing to spend a few hours every three months pulling the issues together, you could make a valuable contribution to helping share the latest news and information about packet radio developments. TAPR has a regular crew of contributors so finding material is never a challenge!

If you're interested, please contact Bdale Garbee, N3EUA, in care of the TAPR Office.

NEW and RENEWAL MEMBERSHIP APPLICATION

Tucson Amateur Packet Radio Corporation
PO Box 12925, Tucson, AZ 85732

Name: _____

Call License
Sign: _____ Class: _____

Address: _____

City & ZIP
State: _____ Code: _____

Home Work
Phone: _____ Phone: _____

If you wish to have any of the above information deleted from publication in a membership list, please indicate which items you wish suppressed:

I hereby apply for membership in TAPR. I enclose one year's membership dues (\$15 US, \$18 Canada/Mexico, \$25 Outside North America).

Signature: _____

The Tucson Amateur Packet Radio Corporation is a non-profit, scientific research and development corporation. TAPR is chartered in the State of Arizona for the purpose of designing and developing new systems for packet radio communication in the Amateur Radio Service, and for freely disseminating information required during and obtained from such research.

The officers of the Tucson Amateur Packet Radio Corporation are:

Andy Freeborn, N0CCZ President
Peter Eaton, WB9FLW Executive Vice President
Bdale Garbee, N3EUA Secretary/Treasurer

The *Packet Status Register* is the official publication of the Tucson Amateur Packet Radio Corporation. Second-class postage paid at Tucson, AZ and additional mailing offices. **POSTMASTER: Please send address changes to TAPR at the address shown below.** Explicit permission is granted to reproduce any material appearing herein, providing credit is given to both the author and TAPR.

**TAPR Membership &
PSR Subscription Mailing Address:**
Tucson Amateur Packet Radio Corp.
PO Box 12925
Tucson, AZ 85732
Phone: (602) 323-1710

PSR Editorial (Only) Address:
Scott Loftesness, W3VS
Packet Status Register Editor
16440 Rustling Oak Court
Morgan Hill, CA 95037

Your Packet Status Register - June 1989

**Tucson Amateur Packet Radio Corp.
PO Box 12925
Tucson, AZ 85732**

*Second Class Permit Pending
Tucson, AZ*

FIRST CLASS MAIL



Check your address label for membership expiration date. Your renewal is important!