PSR Quarterly

TUCSON AMATEUR PACKET RADIO CORPORATION

PRESIDENTS CORNER

Lyle Johnson, WA7GXD

Looking back on 1985...

The past year witnessed a tremendous growth in the ranks of Amateur packeteers. At the beginning of 1985, we estimated about 2,500 TNCs of all types had been placed in the field. At the close of the year that number is closer to 11,000!

In April, TAPR ceased production of the TNC 1 kit. This unit is, however, still available through Heathkit as the HD-4040 kit, as limited parts sets through Applied Digital Technology and as a wired and tested unit through AEA as the PKT-1.

Also in April, TAPR unveiled the TNC 2. Representing a price/performance breakthrough, 1200 kits were sold in the twelve weeks from late August through late November, 1985. TAPR has since discontinued production of this popular unit!

However, through licensing arrangements, the TNC 2 is now available from AEA as the PK-80 (wired and tested), from GLB as the TNC-2A (kit), from MFJ as the MFJ 1270 (wired and tested) and from Pac-Comm as the TNC-200 (kit or wired). Unlike the TNC 1 license, TAPR will be receiving royalty payments for the next two years based on production of these licensed units. Which set the stage for a third major development in 1985.

TAPR has left the TNC market. Now, if a manufacturer comes out with a super chip that will enable us to design a $25 TNC, we probably will, but we have no plans for producing another TNC and none are in the works at the time of this writing.

1985 also saw TAPR exit the world of red ink and embrace the world of black ink! In other words, TAPR has settled all outstanding debts and has sufficient cash reserves to carry us through 1986.

At the ARRL Ad Hoc Digital Committee meeting in December, KA9Q revealed his source code for the TCP protocol and announced his IP protocol implementation was functional, but it (IP) needed considerable fleshing out. This means that a level three and a level four protocol will soon be in testing on Amateur channels.

Not to be outdone, N2WX has had prototype AX25/AX75 Level Three code running on some TNC 2s in Florida.

SAREX 2 – PACKET RADIO FROM THE SHUTTLE

TOM CLARK, W3IWI
ERIC ROSENBERG, WA6YBT

With the unqualified success of the amateur radio experiments and operation by Owen Garriott, W5LFL, Tony England, WOORE, and the three hams on the German Spacelab D-1, it was only a matter of time before a new amateur radio experiment was devised. FM voice and SSTV modes had been tried, packet radio is next!

This logical growth began last summer when Bob Bruninga (WB4APR), Ron Parise (WA4SIR) and Tom Clark (W3IWI) began looking at the possibilities and by September had formed a group that came up with a plan to put a packet radio station on a Space Shuttle mission. Dick Daniels, W4PUJ, AMSAT's NASA liaison, knew the inner workings of both the Agency and the mechanics of the Shuttle program and Bill Tynan, W3XO who was coordinating AMSAT's manned-flight operations.

Early on, it was decided that official support from AMSAT and/or the ARRL would be imperative. Both organizations had been successful in persuading NASA to let Owen Garriott take ham radio equipment with him on STS-9 in the Fall of 1983, and Tony England two years later. Without their support, the fate of the project lay in the balance. Fortunately, both organizations gave their support to the project. The AMSAT board approved the proposal and modest budget at their annual meeting in early November, where AMSAT President Vern Riportella, WA2LQQ, made the official presentation of the proposal to Dick Daniels. W3IWI presented the concept to AMSAT's Board of Directors and (since there were no other volunteers) was promptly given the task of making it all happen.

When the news had gotten out that Ron was scheduled to fly on the 61-E "ASTRO-1" mission in March 1986, the project suddenly took on a new life. Specialists from across the country were recruited to develop the equipment and methods needed to make Ron's packet radio experiment a reality. From Florida, Howie Goldstein (N2WX, developer of the TNC2 software) was recruited to write new code that would support three separate beacons while allowing stations to connect to the the SAREX stations when the shuttle was in view. At the same time, Bob McGwier (N4HY) in Alabama was tasked with working on a means to run a WORLI-style

Thus, we are on the brink of testing Networking protocols in Amateur packet radio!

TAPR has also been soliciting inputs for, and doing the design of, a Networking Controller (NNC). December saw the second revision of the PC boards for that design sprout ICs. Initial tests are very promising.

...and ahead to 1986.

As low-cost hardware enters the marketplace, more and more folks will be joining the swelling packet ranks. I expect our numbers to at least double, so we may have as many as 25,000 packeteers by the end of 1986!

The NNC should be undergoing Alpha phase (software development) starting in late January or February. Beta testing (thrashing the software and hardware in real packet environments) should commence during the second quarter, leading to general availability sometime in the summer. Networking should become a limited reality!

Behind the scenes work on a project called SAREX2 may bring packet to the cockpit during one or more Shuttle mission sin 1986. This will provide a testbed for some store-and-forward experiments of the PACSAT variety; more importantly, it will provide global exposure to Amateur packet radio. This could lead to increased interest in Amateur radio and help bring badly needed fresh blood into our ranks.

In mid-summer, a pair of Amateur satellites carrying packet may be launched into earth orbit. One, the Japanese JAS-1 will carry a packet-only store-and-forward experiment using a 2-meter uplink and a 70 cm downlink. The other, Phase 3-C, will carry an experiment called RUDAK, which is more like a high-altitude (40,000 km!) digipeater.

The upside of both these satellites is the packet capability. The downside is that neither can use an off-the-shelf TNC without an external modem.

So, expect TAPR to have available reasonably priced kits that will plug into the external modem connector on the TNC 1/TNC 2 and provide the modulation modes and speeds necessary for proper operation on these satellites.

GLB has announced a packet RADIO that we hope to help test in January. This will be the first radio available to Amateurs specifically designed to handle packet communications, and should work at 9600 baud, perhaps up to 19,200 baud.

AMRAD has announced work on a radio for packet that may operate at speeds of 9600 baud to perhaps as fast as 56 kbps.

Investigation into economical, spectrally efficient modems for weak signal work (HF and OSCAR linear transponders) will be ongoing in 1986.

Expect work on higher level protocols to accelerate in 1986 as well. WA4FIB presented a specification for Level 5 (Session) at the Southnet II conference held in Atlanta in November. NC4E presented a standardized approach for message handling on packet that is compatible with current CW/RTTY/PHONE traffic procedures used by NTS.

RUNNING A TNC-2 ON 7.5 VOLTS
Chuck Green, NOADI

TAPR engineers recently built three TNC-2's for use aboard the Space Shuttle.One of the special requirements of these TNC's was that they run on +7.5 volts DC. It seems likely that this capability would be of interest to quite a few people so a list of the things changed to satisfy the 7.5 volt requirement follow.

POSITIVE 5 VOLT SUPPLY

A LM2940 is used for Q3. This voltage regulator has a much lower minimum forward voltage drop (a few hundred millivolts) than the original 7805.A +4.9 volt output can be maintained with an input voltage of about 5.4 volts.

NEGATIVE 5 VOLT SUPPLY

The negative voltage generator is the part of the TNC which requires the highest input voltage. Three things were done to minimize this requirement. (You should have already changed C10 to at least 47 mfd to avoid potential problems. Also, note two errors on page 3 of the schematic: C9 and C11 are both .1 mfd.)

1) 1N5818 Shottky diodes were used for CR2-5. They have a forward voltage drop of only about 300 millivolts.

2) R1 was replaced by a 10 uh inductor (same as L2 and L3). It would probably work just as well to use a piece of wire.

3) A 1N751 was used for CR6. This part has a Zener voltage of 5.1; the original 1N754 had a Zener voltage of 6.8 volts. A -4.9 volt output from Q2 should be available with an input voltage of about 7.2 volts.

BATTERY BACKED-UP MEMORY PROTECTION CIRCUIT

The threshold voltage at which the RAM is disabled needs to be lowered. This was done by using a 1N752 for CR8. This part has a Zener voltage of 5.6; the original 1N754 had a Zener voltage of 6.8 volts. It was also necessary to change R9 and R10 to 2 K (the originals were 10K). The 1N752 needs the additional current flowing through it to function properly. (It may be that some original 1N754's could also benefit from this current increase.) With these changes, the RAM IC's will be disabled when the input voltage drops to about 6.0 volts.

The result of these changes is a TNC that will run reliably on five flashlight batteries ("C" cells) instead of eight. Does this give you any ideas about a completely portable (HT radio and lap computer) Packet station for demonstrations or emergency use? Also, please note that TAPR does NOT stock these parts; you will have to find them locally.

It is time to vote for Directors once again. Please locate the list of candidates in this PSR, then select five (5) and mail your ballot in to the TAPR PO Box. Do it today!

bulletin board on the Tandy Model 100 computer donated to the project by Radio Shack; Jack Colson (W3TMZ) agreed to the task of flight-hardening the Model 100.

Farther west, Lyle Johnson (WA7GXD), Chuck Green (N0ADI) and the TAPR organization in Arizona agreed to provide flight-hardened TNC-2's. The clubs at NASA/GSFC (under Frank Bauer, KA3HDO and Dick Kutz, KS3Q) and NASA/JSC (under Lew McFadin, W5DID and Dick Fenner, W5AVO) agreed to assist in integration, testing and greasing the skids at NASA. Kai Siwiak, KE4PT and the Motorola ARC agreed to support adapting the Motorola MX- series HT's (as used on previous flights) to support this mission. Neil, KA3DBK and Carlos, KA3KIW agreed to help bolt everything together.

While all of the hardware and software development was going on, W3IWI was marshalling the forces on packet, E-Mail, and telephone for suggestions on how the operation would actually work. Tom was wondering what the beacons would actually transmit, how Ron would be able to log all the stations calling and working him. At the root of all of the discussions was the underlying question, "Just how many packeteers are there?"; packet radio's west-coast pundit NK6K came through with his predictions that there would be approximately 10,000 TNCs out there by the March flight date.

Thus SAREX2 was born. SAREX is an acronym for "Shuttle Amateur Radio EXperiment"; SAREX1 was W0ORE's SSTV experiment flown on Spacelab-1.

The timeline developed for meeting the March flight date was really tight! It was necessary to have flight-worthy hardware and software ready for final testing in mid-January. And to be included in the testing, approval had to be received from NASA. Based on W3IWI's constant prodding the hardware and software modules were readied on schedule for final (it better work the first time!) integration over the Christmas holidays.

In early December, things took a turn for the worse. NASA's Administrator, Jim Beggs was indicted on charges of conspiring to defraud the Army while he was a corporate officer with General Dynamics, and William Graham was named acting Administrator of NASA. This seemingly disjoint happening rippled thru the SAREX2 approval since the "prime sponsor" of the shuttle amateur radio experiments is NASA Headquarters Office of Public Affairs. Needless to say, the Beggs/Graham situation kept them somewhat preoccupied at a very critical time for flying in March!

Despite this setback, the SAREX2 team continues to make the hardware ready for the earliest possible flight. We can see 3 possibilities in the next year with WA4SIR's 2nd ASTRO flight and probable flights by W0ORE and W5LFL.

SOME MORE DETAILS

The TNC2 to be flown on SAREX2 is a Rev.2 unit board with only minor changes from normal production units:

- it has been populated with Mil-spec "hi-rel" parts
- it doesn't use the low-cost DB25, power and DIN connectors
- the power supply has been changed to run from +7.5 VDC
- mechanical mounting is different
- the ROM has special code.

WORKING THE SHUTTLE

The last item, SAREX2 TNC2 software, is worth some further discussion. We have seen on previous flights that the astronauts have precious little time to devote to amateur radio, and yet everybody want to work the shuttle. To solve this dilemma, we have taken a page from the Soviet RS- series by including a "ROBOT" automatic QSO machine resident in the TNC2 code. The concept is simple -- the ROBOT is able to work you, give you a serial number, and enter the QSO into the log. A successful QSO requires the following steps:

(1) You send a CONNECT to the ROBOT
(2) The ROBOT sends a connect acknowledge and a brief exchange with your serial number and initiates a DISCONNECT
(3) You acknowledge the data and the DISCONNECT

All serious VHFers know that a successful QSO requires the exchange of some information each way and an acknowledgement to constitute a QSO. This scheme meets that requirement. The ROBOT didn't know your call until it received the connect request. You didn't know the serial number, and you have to acknowledge its receipt in order to have the QSO count. Unless all 3 steps are met, no QSO has been made. Here is a sample ROBOT QSO between W3IWI-1 and the W3IWI-5 SAREX2 ROBOT test/ demo station on the air in Washington; the #04EF is the (hex) QSO serial number:

```
cmd: CONNECT W3IWI-5
W3IWI-5>W3IWI-1:
73 de AMSAT/GARC
*** DISCONNECTED
```

The next special feature of the SAREX2 TNC2 code are the log beacons. We wanted to avoid the necessity of storing the logs on the shuttle and decided to down-link them periodically using two special beacons, HEARD and WORKED; here are some sample beacons from the W3IWI-5 test site:

```
W3IWI-5>HEARD:
0AD6 W3GXT-5 N3AKP N9CG K3VPZ W3IWI-1 W3IWI-6
K2IZS-1 W3ITM W3IWI WB4APR-5 KS3Q WB6RQN N8FJB
KA4USE-1 KA3KIW KA3DBK-1 KA5KVX K4HWG W3VD-5
WA5WTF KE4TZ W3HXN K2IZS KB3DE W4CQI
```

```
W3IWI-5>WORKED:
W3IWI-1/04EF  W3GXT/04E4   K1HTV/04E3   WB3LKB/04D7
W3IWI/04D5    W9BVD/04D0   W3VH/04CD    KA3HDO/04C8
K3LLL/04C7    K3VPZ/04C5   W3TMZ/04BE   N3BRQ/04BB
N3CHX/04AE    N3BDW/04A9   KA3NUA/04A2  KA5KVX/047E
W3IUI/047C
```

In the HEARD beacon, the most recent 25 calls heard by the TNC2 are sent in the order they have been heard. New calls appear at the top of the list, old calls are pushed off the bottom. The HEARD beacon begins with a (hex) beacon serial number (0AD6 in this example) so that logs can be pieced together on the ground.

The WORKED beacon lists those who were lucky enough to get a serial number and an acknowledgement thru. If a serial number is sent but no acknowledgement is received, the serial number is discarded. If you work the ROBOT and your call is already in the WORKED list, your earliest QSO serial number is retained. The 17 most recent unique 2-way QSO's are listed in order. Older QSO's are pushed off the bottom of the list. If you were on the list before, and your call was pushed off the end, your next QSO puts you back onto the list. Note that the W3IWI-1 QSO shown

Continued from previous page.

earlier re-established W3IWI-1 at the head of the WORKED list.

The ROBOT code retains all the normal TNC2 features and makes good use of the latest multi-connect TNC2 features; up to 9 QSO's can be in progress at any given time. The code has been on the air at two "beta" test sites (N2WX in Florida, W3IWI-5 in Maryland) since late November. More recently, two other temporary sites were used: W3IWI set up camp in San Francisco while on a business trip and some 200 connects where logged. Meanwhile WA0JFS, W0RPK and the Central Iowa Technical Society took their prototype for an hour long free-for-all in an airplane at 8,000 feet over Central Iowa when more than 500 successful connects were recorded. In addition, on December 21st, the FADCA group (under Henry, WA4HXZ) flew a TNC 2 with SAREX2 (release 1.1.2a), and a Motorola HT up the middle of the state. 899 ROBOT QSOs were made, and no significant problems were reported. The reception from the packet community has been overwhelming. SAREX2 ROMs have been shipped to Houston, Auburn and Los Angeles for additional testing. The Houston site is particularly important so that we can begin indoctrinating W0ORE, W5LFL and the shuttle hardware people about the wonders of packet radio.

OTHER SOFTWARE AND MODES

In addition to the ROBOT, there are several other interesting features in SAREX2. In addition to a "normal" beacon (which is sent along with the HEARD and WORKED beacons), N2WX and W3IWI devised a scheme for a "meta- beacon" to be sent every 4 minutes; this involves tricking the TNC2 into thinking it is permanently connected to a user (like WORLD in the following example), setting RETRY to zero, and filling the outgoing transmit buffer with up to 7 frames of 256 bytes each (about 1.7 kbytes total). This will be loaded by the astronaut with informative text (like mission status info, greetings to the world, etc.) using the Model 100 when he has time.

The Model 100 will also have "mini-WORLI" BBS software to support special store-and-forward demonstration experiments a la PACSAT; these experiments will be on a scheduled basis with pre-selected ground stations. If time permits during the flight, the Model 100 also has split-screen terminal emulator software for real-time QSO's.
The TNC2 and a power conditioning module will be housed in a 9" x 12" x 2" shielded box which will be attached to the base of the Model 100 with Velcro, resulting in a 9" x 12" x 4" porta-packet station. Power will be derived from the Shuttle's +28VDC bus with an isolated DC/DC power converter which produces +7.5 VDC; the TNC2, Model 100 and Motorola MX- series radio have all been configured to run on +7.5VDC. Also inside this box is extensive RFI shielding to insure that the packet hardware will not QRM the MX-series radio or (God forbid!) the shuttle itself. The Model 100 has been modified internally to improve its shielding too; however the Model 100 can be left off during normal "ROBOT" operations to minimize power drain and RFI.

The MX-series radio is the same as was used by W5LFL and W0ORE and operates on 2 meter FM; it is anticipated that most operations will have the shuttle transmitting on 145.55 MHz and listening on 144.95 MHz (with a standard 600 kHz split). The antenna will also be the same window-mount antenna used on earlier missions.

USING THE RF DCD FEATURE OF THE TNC-2

Lyle Johnson, WA7GXD

Ever wondered what the RF DCD input does on your TNC 2? Or how to apply it in your packet station? This article explains the oft-alluded to but never exposed feature!

RF DCD

Most packet operation on VHF or HF is done on "packet-only" (or "packet-mostly") channels. However, you may desire to operate on a shared-mode channel (such as a voice or packet full-duplex repeater). In such a case, it is often considered bad operating procedure to send a packet in the middle of someone's voice traffic!

There are at least two solutions to this problem.

The first is to monitor the channel with another radio, or with the speaker enabled on the radio you use for packet. This should ALWAYS be done in any event. It is then a simple matter to strike the carriage-return key on your keyboard at the end of the voice transmission and presto! your packet Brrraaaaap (and the other station's return ack) will "time slice" with the voice traffic. They hear a strange "cou extended range packet QSO via the repeater.

However, there is an automatic way to share other modes. If your rig has a squelch output, you can interface it to the TNC 2's RF DCD input on the radio connector (J2 pin 5 -- see TNC 2 System Manual, chapter 3 page 1). Activating this input will hold off your packet transmissions in the same manner as the normal DCD circuitry in TNC 2's modem holds off packet transmissions in the presence of incoming data. Naturally, FULLDUP must be OFF for this feature to operate (see the FULL-DUP command in the TNC 2 System Manual, chapter 6).

The RF DCD simply requires a ground to activate it. Please refer to your TNC 2 schematic (Rev 1 or Rev 2), sheet 2 of 3, for the following discussion.

How RF-DCD Works

The RF DCD input pin (J2 pin 5) is pulled up to +5 volts by 10k resistor R38. It is decoupled via diode CR13, so the pin can be pulled up to about +30 volts without damage. When grounded, something under 500 uA will flow (1/2 mA) through pin 5 and the externally applied ground. Schmitt trigger U9 requires an input voltage below 0.9 volts to trigger. It can have an input leakage current of up to 1 uA, and this current will flow through R42, a 100k resistor, which can drop as much as 0.1 volt. CR13 can drop as much as 0.6 volts, so the external RF DCD input should be pulled to within 0.2 volts of ground. A saturated NPN transistor (2N3904, etc.) can provide this sort of ground (Vce sat) with 1 mA or so of base current, or a VFET can be used. A logic gate output (TTL or CMOS) will also suffice, although the CMOS output is much better, with a TTL output being somewhat marginal.

After interfacing RF DCD to your radio, you simply operate the radio squelched. When activity occurs, the squelch will break, signalling the TNC to refrain from transmitting. When the channel is quiet, the TNC is free to transmit any data it may have.

NNC DEVELOPMENT AND TESTING

As many of you are already aware, the new TAPR Networking Node Controller (NNC) is nearing completion of prototype hardware debugging. The NNC is a four-port packet controller with a large memory area, direct-memory access (DMA) capability for I/O -- and Z-80 software compatibility.

To dispell any rumors, there presently exists NO SOFTWARE for this device. None. Nada. Ayn. Zip. Zero. Effes. Klum. ( )

What we are looking for are volunteers to assist in developing software for this device.

We need low-level, highly-efficient drivers for the I/O. We need an AX.25 Level Two handler that can handle multiple logical and physical channels. We need Level Three and Level Four. We need loaders for uploading software updates to a remotely-sited NNC. We are hoping that there will be early porting of multi-port digipeater code to this unit as well as a WORLI PBBS. We need close coordination of the various aspects of the development. We need... You get the idea.

The hardware should be verified during December. If all goes well (it usually doesn't), we will want to put Alpha units in the hands of developers in late January/early February. Assuming a couple of months to get enough software together to make Beta testing meaningful, we will be looking for Beta testers in the March-April timeframe. Once testing has advanced to the point of reasonable confidence, we will make the units generally available (summer of 1986?).

Now, we are NOT looking for folks who want to be the first kid on the block with a new toy. We need people who are committed to Amateur packet radio and want to help make a meaningful contribution to a very large and difficult task.

And be forewarned. You may slave away for many, many hours, only to have your code not used, or superceded, or... No guarantees.

Coordination is going to be a tough assignment. Without proper coordination, a lot of wheels will spin, and a lot of energy wasted in duplication of efforts. A BBS to swap code modules will be needed. All code will need to be carefully, accurately and exhaustively documented -- by the author!

Developers will need to procure the following:

1) One NNC digital unit - projected cost is $175. This is an NNC with uP, 64k bytes of bbRAM, 32k bytes of EPROM, four HDLC ports, two parallel (centronics compatible) ports, two async ports and one SCSI interface. The SCSI chip may not be included at this price, we are not sure yet, but for the Alpha testers/developers it will be. This unit will be fully assembled and "tested."

2) One NNC Floppy Adapter - projected cost $125. This includes a DMA'ed Floppy Controller that can handle 4 diskette drives. This unit will NOT support 8" drives (lack of 8" support is intentional). The price will include a licensed copy of Z-DOS, a CP/M 2.2 compatible operating system. It will be on 5.25" double-sided 48 tpi diskette format capable of 386k bytes (formatted). If the decision

is made up front to use 96 tpi drives, TAPR will copy the licensed diskette to the denser format and supply the original as well as the high-density copy to the purchaser.

3) A pair of 5.25" floppy drives. Maybe we can do a group purchase of TEAC 55Bs. Figure $150 for this expense. 48tpi or 96tpi are about the same price. 48 tpi yield about 400k formatted bytes; 96 tpi about 800k formatted bytes.

4) A power supply. $50 from surplus sam?

5) One NNC Modem board-projected cost is $150. This is a wired and tested board which includes one 300-baud 2206/2211 modem with tuning indicator and three 1200-baud 2206/2211 modems. We might get this cost down to $125.

Thus, there is a cost of participation that will be a minimum of $450 and may be $650. Add to this the cost of an assembler or compiler...

The assembler that seems to make the most sense is ZAS, from Echelon systems. Again, we can probably do a group purchase or multiple-site license for this project. This assembler supports the extended instruction set of the HD64180 cpu. There is no reason to limit ourselves to the Z80 instruction set (or - yeech - the 8080 subset) for this project. And ZAS is fairly cheap - about $50.

We don't know which C or Pascal compiler will be chosen. It is safe to assume that one will be chosen, so the high-level code can be written in a transportable high-level language (makes for easier testing?) while the interfaces to the hardware can be done in assembly language. Preferably, the compiler will generate Z80 (or 64180) source code for assembly by ZAS. This allows hand-optimization of the compiler output.

By standardizing on the development environment (NNC w/5.25" floppies) and the tools (assembler, compiler(s)), we hope to make it easier for all participants to share their work amongst the group.

It is expected that all code (including source code) developed for this project will be placed in the public domain for non-commercial use. And that TAPR will be given explicit (not exclusive) right to distribute it.

If you have the time and ability and want the chance to make a real contribution to Amateur packet radio networking development, please leave a message on DRNET or write the TAPR office. We will put you on file and notify you when we are ready to get started with Alpha test or Beta test (as you indicate to us).

For Alpha test, we need developers. Committed developers. People who really understand software design, hardware/software interaction, protocol implementation, code size/speed tradeoffs, data structures and myriad other facets of software design. And of course, understand networking...

For Beta test we need testers. People who are in a real packet environment, who have a good site that will get plenty of exercise on the air, who have the time and commitment to submit detailed reports of what works and what doesn't. This isn't a "be the first person on your block to own an NNC" contest; it is going to require work.

IMPORTANT: TO ALL TNC-2 OWNERS

Herewith are a couple of potential problem areas on TNC 2 along with suggested fixes.

-5 volt supply

There is a potential problem with all TNC 2s. Capacitor C10, at the output of the -5 volt regulator transistor, Q4, is specified as 10 uF.

This value is marginal. In a very small number of TNC 2s, latchup of the MF10 filter has occured due to power supply turn-on characteristics.

Please change this capacitor to either 47 uF or 100 uF. This will ensure that the power supplies turn on properly and reduce the possibility of future latch-up. If you cannot obtain such a capacitor locally, write to the TAPR office and we will supply one free of charge.

NOTE: At least one person has reported connecting a shottky diode across C10, with anode to -5 and cathode to ground, as a satisfactory solution to this same problem.

7 kHz-spaced HF RFI

Some people have noted a raspy-sounding noise every 7 kHz or so on HF when the HF rig is located in close proximity to a TNC 2.

The noise appears to be generated by the charge pump, U2, according to Bill, N9CX, in Indianapolis. That group offered the following solution, which TAPR has tested and verified.

If you experience this noise, connect a 0.1 uF capacitor between U2 pins 3 and 7, and another 0.1 uF capacitor between U2 pins 11 and 7. This will bypass the voltage references used inside the chip. In addition, grounding the chassis of the TNC 2 via a connection to one of the end plate screws should prove helpful.

Corrections to TNC 2 Rev 1 Upgrade

The last PSR Quarterly contained an article dealing with upgrading a Rev 1 TNC 2 to the Rev 2 level. Unfortunately, two steps were left out. Referring to the memory modifications on Page 9 of that issue, add the following two steps:

7A: Cut the trace from U24 pin 26 to U23 pin 26.

8A: Add a jumpoer from U24 pin 26 to U24 pin 28.

NOTE: A complete update kit with expanded memory and corrected instructions has been sent free-of-charge to every TNC 2 Rev 1 kit owner in our records. If you have a TNC 2 Rev 1 and haven't received an upgrade kit, contact the TAPR office immediately. Thank you!

Potential Problem with Rev 2 Boards

R98 is supplied as a 4.7k ohm resistor. However, there is a chance that the SIO may cause a glitch in the reception of packets with this value. Please change the resistor to 100k ohms for best operation.

This resistor was added in the Rev 2 design to accomodate possible future software modes such as AMTOR; Rev 1 boards do not include R98.

TAPR ANNUAL MEETING

The Tucson Amateur Packet Radio Annual Meeting will be held Saturday, February 8th, 1986 at the Embassy Suite Hotel located at the entrance to Tucson International Airport. While the name has changed, this is the same physical location as last year's meeting.

The meeting will last from 9 AM until 5 PM, with a catered lunch. Price for the lunch will be $7.50. An informal dinner will be held at a local restaurant (steak house or mexican - to be determined) on Saturday evening.

There are no special rates at any hotels in Tucson for this meeting. For those of you who are on a tight budget, or just want to rub shoulders with other packeteers, the traditional hostelry is called the Allstar Inn and the 'phone number there is 622-4614. The normal rate there is $25.95/night.

If you come to town for the meeting, you can relax Sunday and take in the sights at the Arizona-Sonora Desert Museum or other famous attractions.

Notice to Directors: The Board of Directors meeting will be held Sunday, February 9th. Please plan on being available until at least 5 PM Sunday evening for this meeting.

TAPR TNC-1 MANUAL SALE

TAPR is closing out the remaining 30 copies of the TNC 1 manual. If you own a heath HD4040 or AEA PKT-1, there is valuable information in this manual that is not contained in the documentation supplied with your TNC.

Normally a bargain at $20, TAPR is closing them out at the incredible price of only $7.00! (and this price includes surface shipping within the continental United States!)

Get yours now!

NNC Development and Testing

What do we mean by commitment?

Consider a plate of bacon and eggs. The chicken was actively involved; the pig was committed.

If you are a capable packeteer, committed to assist in networking development, please provide us with the following:

Full Name.
Amateur Callsign.
Mailing Address.
Daytime telephone number.
Evening telephone number.
Alpha or Beta test.
TAPR membership number (if applicable).
Specific areas of expertise that you wish to make available to this project (low level interface/high level protocol implementation/documentation/testing/etc.)

Thank you for you help. Happy packeting!

TCP PROGRAMERS MANUAL
Phil R. Karn, KA9Q

My TCP implementation work for amateur packet radio has reached the point where I can describe the interface provided to the application by TCP. I have also written a UDP (User Datagram Protocol); however, this is not at the same level of maturity as TCP and is therefore more subject to change. This note is meant primarily as "advance information" to any implementers considering writing applications.

To review the purpose of TCP: it supports a reliable, sequenced, byte stream "connection" on an end-to-end basis. It fits roughly at the Transport layer (level 4) of the OSI model. Since a single TCP module supports multiple connections through the use of port numbers, it also provides Session layer (level 5) functionality without the need for a distinct protocol. (Or it makes the session layer unnecessary, depending on your point of view). This package is written as a "module" intended to be compiled and linked with the application(s) so that they can be run as one program on the same machine. This greatly simplifies the user/TCP interface, since it becomes just a set of internal subroutine calls on a single machine. Reliability is much greater, since a hardware failure that kills TCP will likely take any applications with it anyway. Only IP datagrams flow out of the machine across hardware interfaces (such as asynch RS-232 ports or whatever else is available) so hardware flow control or complicated host/front-end protocols are unnecessary.

A TCP connection is uniquely specified by the concatenation of source and destination "sockets". In turn, a socket is the concatenation of a host address (a 32-bit integer) and a TCP port (a 16-bit integer), defined by the C structure

```
struct socket {
long address;/* 32-bit IP addr */
short port;/* 16-bit TCP port */
};
```

Therefore it is possible to have several distinct connections established at the same time to a single port on a given machine, as long as the source sockets are distinct. Port numbers are used either through mutual agreement, or more commonly when a "standard" service is involved, a "well known port" number. For example, to obtain standard remote login service (known as "telnet") one initiates a connection to TCP port 23; to send mail using the Simple Mail Transfer Protocol (SMTP) one talks to port 25. ARPA maintains port number lists and periodically publishes them. They will also assign port numbers to a new application on request if it appears to be of general interest.

TCP connections are best modeled as a pair of one-way paths (one in each direction) rather than as a single full-duplex path. Station A may close its path to station B leaving the reverse path from B to A unaffected. B may continue to send data to A indefinitely until it too closes its half of the connection. This is known as "graceful close" and can greatly simplify an application.

My TCP code supports five basic operations on a connection: open, send, receive, close and delete. A sixth, tcp_state(), is provided mainly for debugging. They are summarized in the following section in the form of C declarations and descriptions of each argument.

```
int net_error;
```

This global variable is used to indicate the specific cause of an error in one of the TCP or UDP functions. All functions returning integers (i.e., all except open_tcp) return -1 in the event of an error, and net_error should be examined to determine the cause. The possible errors are defined as constants in a header file.

```
/* Open a TCP connection */
struct tcb *
open_tcp(lsocket,fsocket,active,
notify,tos)
struct socket *lsocket,*fsocket;
int active;
void (*notify)();
char tos;
```

"lsocket" and "fsocket" are pointers to the local and foreign sockets, respectively.

"active" is 0 for a "passive" open (one in the TCP LISTEN state). A passive open does not cause any packets to be sent, but enables TCP to accept a subsequent active open from another TCP. If a specific foreign socket is passed to a passive open, then connect requests from all other foreign sockets will be rejected. If the foreign socket fields are set to zero, then connect requests from any foreign socket will be accepted. If "active" is 1, TCP will initiate a connection to a remote socket that must previously have been created in the LISTEN state. The foreign socket must be completely specified in an active open.

"notify" is an optional receive "upcall" mechanism, useful when running in a non operating system environment. If "notify" is non-zero, it is taken as the address of a function to be called whenever a "significant" amount of data arrives. This user-provided function may then invoke recv_tcp() to obtain the incoming data.

"tos" is the Internet "type of service" field, consisting of precedence and class of service parameters. There are 8 levels of precedence, with the bottom 6 defined by the military as Routine, Priority, Immediate, Flash, Flash Override and CRITICAL. (Two more are available for internal network functions). For amateur use we can use the lower four as Routine, Welfare, Priority and Emergency. Three more bits specify class of service, indicating that especially high reliability, high throughput or low delay is needed for this connection. The entire TOS field is passed along to IP in each datagram and is interpreted by each IP gateway (packet switch) in the route. The precedence value actually used is the higher of those specified in the two tcp_open() calls.

open_tcp() returns a pointer to an internal Transmission Control Block ("tcb"). This "magic cookie" must be passed back as the first argument to all other TCP calls. In event of error, the NULL pointer (0) is returned.

The only limit on the number of TCBs that may exist at any time (i.e., the number of simultaneous connections) is the amount of free memory on the machine. Each TCB on a 16-bit processor takes up about 129 bytes; additional memory is consumed and freed dynamically as needed to buffer send and receive data. Deleting a TCB (see the delete_tcb() call) reclaims its space.

```
/* Send data on a TCP connection */
int
send_tcp(tcb,data,cnt)
struct tcb *tcb;
char *data;
unsigned cnt;
```

"tcb" is the pointer returned by the open_tcp()
call. "data" points to the user's buffer, and
"cnt" specifies how long it is. The number of
bytes actually queued for transmission is return-
ed; it will equal "cnt" unless some sort of error
occurs, such as lack of memory for buffering. The
data is copied into an internal queue, a transmis-
sion is attempted, and the function returns so
that the user may immediately reuse his buffer.
TCP uses positive acknowledgments and retransmis-
sion to ensure in-order delivery, but this is
largely invisible to the user. TCP enforces no
limit on how much data can be queued for transmis-
sion, so the user should be careful not to run the
system out of free memory. (This is something else
that requires a multitasking kernel to do right).

```
/* Receive data on a TCP connection */
int
recv_tcp(tcb,data,cnt)
struct tcb *tcb;
char *data;
unsigned cnt;
```

The arguments to recv_tcp() are identical to those
of send_tcp(), except that any data on the connec-
tion's receive queue is placed in the user's buf-
fer, up to a maximum of "cnt" bytes. The actual
number of bytes received (the lesser of "cnt" and
the number pending on the receive queue) is re-
turned. Since this TCP module cannot assume the
presence of sleep/wakeup primitives provided by an
underlying operating system, recv_tcp() is cur-
rently designed to return -1 with net_error set to
EWOULDBLK if no incoming data is pending. The
"notify" feature on open_tcp() is provided to
eliminate the need for constant polling of the
recv_tcp() function; whenever TCP calls the notify
function, it guarantees that recv_tcp() will not
return -1. (Technical note: "notify" is called
whenever a PUSH or FIN bit is seen in an incoming
segment, or if the receive window fills. It is
also called before an ACK is sent back to the
remote TCP, in order to give the user an opportun-
ity to piggyback any data in response.)

When the remote TCP closes its half of the connec-
tion and all prior incoming data has been read by
the local user, subsequent calls to recv_tcp()
return 0 rather than -1 as an "end of transmis-
sion" indicator.

```
/* Close a TCP connection */
close_tcp(tcb)
struct tcb *tcb;
```

This tells TCP that the local user has no more
data to send. However, the remote TCP may continue
to send data indefinitely to the local user, until
the remote user also does a close_tcp(). An at-
tempt to send data after a close_tcp() is an
error.

```
/* Delete a TCP connection */
delete_tcp(tcb)
struct tcb *tcb;
```

When the connection has been closed in both con-
nections and all incoming data has been read, this
call is made to cause TCP to reclaim the space

taken up by the TCP control block. Any unread
incoming data is lost.

```
/* Dump a TCP connection state */
tcp_state(tcb)
struct tcb *tcb;
```

This debugging call prints an ASCII-formatted dump
of the TCP connection state on the terminal. You
need a copy of the TCP specification (ARPA RFC 793
or MIL-STD-1778) to interpret most of the numbers.

Well, that's it. Constructive comments on this
interface are welcomed.

BEGINNER'S CORNER

Using the IBM PCjr on Packet
Lyle Johnson, WA7GXD

In addition to the Commodore computers (see else
where in this PSR Quarterly), we have had several
inquiries regarding the use of the IBM PCjr with
Amateur packet radio.

While there are many programs available for the
IBM PC family of computers for telecommunications,
most will not work properly with the PCjr.

The problem is simple: the PCjr does not use
direct memory access (DMA) techniques with its
disk drive. As a result, most communications
programs written for the PC will lose data if run
on a PCjr with a disk capture file open.

The solution is equally simple: tell the TNC to
stop sending data when before doing a disk access,
then allow data from the TNC after the disk write
is done.

Simple? Yes, but I haven't been made aware of any
program that does this! Further complicating
things, the PCjr uses the 8088 cpu for software
decoding of the wireless keyboard (packet radio
using the software approach?). This means that
the data rate between the TNC and the PCjr should
not exceed 1200 baud or characters may be lost
from the serial port if you happen to type at the
keyboard at the same time a character is coming in
from the port.

A program is presented here that solves these
problems. It politely requests the TNC to stop
shipping data before a disk access is done. It
pauses in case the TNC has a character or two in
its UART (6551 for TNC 1, SIO/0 for TNC 2) that it
wants to ship. The disk write is then done.
After the disk write, the TNC is allowed to con
tinue dumping data to the PCjr.

The program is written in BASIC. As such, it is
slow. The program is also a bit simple-minded.
It does not handle the <BS> character properly on
display, so be sure to run the TNC with BKONDEL
OFF. Correcting this minor annoyance is left as
an exercise for the reader...

In case this program is used with TNC 1, only
software flow control (X-ON/X-OFF) is used between
the TNC and the PCjr. And the flow control is
only implemented for data from the TNC to the
PCjr. There is no file transfer provision from
the PCjr to the TNC; hence there is no flow con
trol implemented in that direction. You guessed
it, an exercise for the reader!

Herewith is the program. Some notes follow the
listing, so don't type it in until you have fin
ished reading this prose!

```
100 SCREEN 0,0:WIDTH 80
120 KEY OFF:CLS:CLOSE
140 DEFINT A-Z
160 FALSE=0:TRUE= NOT FALSE
180 XOFF$=CHR$(19):XON$=CHR$(17)
200 OPEN "com1:1200,e,7" AS #1
220 OPEN "scrn:" FOR OUTPUT AS #2
240 LOGGING=FALSE
260 LOCATE ,,1
280 PAUSE=FALSE
300 DISK$=""
320 ON COM(1) GOSUB 2000:COM(1)N
340 KEY(1) ON:KEY(2) ON:KEY(3) ON:KEY(10) ON
360 ON KEY(1) GOSUB 3000
380 ON KEY(2) GOSUB 4000
400 ON KEY(3) GOSUB 5000
420 ON KEY(10) GOSUB 6000
440 ON ERROR GOTO 9999
460 B$=INKEY$:IF B$<>"" AND B$<>CHR$(26) THEN
    PRINT#1,B$;
480 IF Z<11000 THEN Z=Z+1
500 IF EOF(1) THEN 460
520 A$=INPUT$(LOC(1),#1)
540 GOSUB 1000
560 LFP=0
580 LFP=INSTR(LFP+1,A$,CHR$(10))
600 IF LFP>0 THEN MID$(A$,LFP,1)=" ":GOTO 580
620 PRINT#2,A$;
640 IF LOC(1)>0 THEN 520
660 IF PAUSE THEN PAUSE=FALSE:PRINT#1,XON$;
680 GOTO 460
1000 IF LOGGING=FALSE THEN RETURN
1020 DISK$=DISK$+A$
1040 IF LEN(DISK$)>127 THEN 1140
1060 IF Z>10000 THEN 1100
1080 RETURN
1100 Z=0
1120 IF DISK$="" THEN 1260
1140 IF PAUSE THEN 1180
1160 PRINT#1,XOFF$;:PAUSE=TRUE
1180 FOR I=1 TO 400:NEXT I
1200 PRINT#3,DISK$;
1220 DISK$="":Z=0
1240 IF LOC(1)<16 AND PAUSE THEN PRINT#1,XON$;
     :PAUSE=FALSE
1260 RETURN
2000 IF LOC(1)>224 THEN PRINT#1,XOFF$;:PAUSE=TRUE
2020 RETURN
3000 X=POS(0):Y=CSRLIN
3020 LOCATE 1,10:PRINT CHR$(201);:FOR I=1 TO 58:
     PRINT CHR$(205);:NEXT I:PRINT CHR$(187)
3040 LOCATE 2,10:PRINT CHR$(186);:LOCATE 2,69:
     PRINT CHR$(186);
3060 LOCATE 3,10:PRINT CHR$(200);:FOR I=1 TO 58:
     PRINT CHR$(205);:NEXT I:PRINT CHR$(188)
3080 LOCATE 2,11:PRINT "    Name of Logging File: "
3100 LOCATE 2,43:INPUT"",DISKFILE$
3120 OPEN DISKFILE$ FOR OUTPUT AS #3
3140 LOGGING = TRUE
3160 LOCATE 25,1
3180 GOSUB 7000
3200 LOCATE 25,1:PRINT DISKFILE$;
     " now logging incoming data.";
3220 LOCATE Y,X,1
3240 RETURN
4000 IF DISK$<>"" THEN GOSUB 1140
4020 X=POS(0):Y=CSRLIN
4040 LOCATE 25,1
4060 GOSUB 7000
4080 LOCATE 25,1:PRINT DISKFILE$;
     " now being closed.";
4100 CLOSE #3:LOGGING=FALSE
4120 LOCATE 25,1
4140 GOSUB 7000
4160 LOCATE Y,X,1
4180 RETURN
5000 IC%=INP(&H2FB)
5020 IZ%=IC% OR &H40
5040 OUT &H2FB,IZ%
5060 FOR WLUP=1 TO 500:NEXT WLUP
5080 OUT &H2FB,IC%
5100 RETURN
6000 CLOSE:CLS:KEY ON:STOP
7000 FOR I=1 TO 79:PRINT" ";:NEXT I:RETURN
9999 PRINT CHR$(7);"Error Number";ERR:CLOSE:END
```

Line 3080 has 9 spaces between the first " and the
word Enter, and 28 spaces between the : and the
closing ".

Function keys are used as follows:

Fn 1 is for opening a log file. You will be
    prompted for a filename.

Fn 2 is for closing the file opened by Fn 1.

Fn 3 is for sending a BREAK character to the TNC
    (useful for escaping from transparent mode to
    command mode).

Fn10 will close all files and exit the program
    (returns you to BASIC so you can modify the
    program some more!).

The backspace key on the PCjr will echo as "\" if
BKONDEL is OFF. If it is on, you will see two
graphics characters with an intervening space.
Run this program with BKONDEL OFF!

The <LF> (linefeed) character is translated to a
blank space to avoid a double-spaced screen (lines
560-600). Change the TNC command AUTOLF to OFF
(default is ON).

If you want to try this program on a standard PC
change the &H2FB occurances in subroutine 5000 to
&H3FB to use the COM1: serial port on the PC.
Another popular MS-DOS computer without DMA is the
Sanyo MBC-55x. I don't know how the Sanyo serial
port works, but if you modify this program and get
it working on the Sanyo, please send us a listing
so we can publish it.

When modifying the program, change line 440 to a
REMARK or debugging may get a bit frustrating.
Also, if the TNC is switched off when you invoke
the program, you will get a "device timeout error"
message. Turn on the TNC before starting the
program!

CONTEST! Send in a modified version of this pro
gram for the PCjr that does the following:

1) Corrects the backspace problem so you can run
    with BKONDEL ON.
2) Allows file dumps from the PCjr to the TNC
    with X-ON/X-OFF flow control allowing the TNC
    to control the data flow from the PCjr.
3) Only eats the first <LF> after a <CR> (this
    will allow proper formatting in case someone
    is sending "RTTY art."
4) Loads up to four buffers from one or more
    disk files so you can send some canned mes
    sages. These messages mapped to function
    keys Fn 5 through Fn 8. A possible use
    includes the ability to send a connect to
    station x via digipeater(s) for connecting to
    your favorite pbbs.

The first three submissions that work (we will test them, so send it on diskette!) will receive two prizes! The first is the return of the disk ette (!) and the second is a new TAPR LSC-10 Packet Station Accessory. The LSC-10 will be unveiled at the TAPR Annual Meeting... (Retail value of the LSC-10 is only $5, so don't get TOO excited!) Everyone else will get a copy of the winning entries on their diskettes IF a proper diskette mailer is used and return postage is affixed.

BEGINNER'S CORNER

Interfacing the Commodore 64 and VIC 20 Computers

Lyle Johnson, WA7GXD

"How do I interface my VIC 20 (or Commodore 64) computer to the TNC 2?" is perhaps the most frequently asked how-to question that appears in the TAPR mailbox. If you have ever asked this question, or are asked for advice by beginning packeteers, read on!

Hardware Required

An RS-232 adapter is necessary for operation of the TNC 2 with the VIC 20 or Commodore 64 computers. While an adapter may be homebrewed, or the RS-232 translation in the TNC 2 can be disabled so you don't need an RS-232 adapter for operation, this article will follow the path of least resistance!

An RS-232 adapter that I have used quite successfully for interfacing the Commodore computers to TNC 1 and TNC 2 is made by Jameco Electronics, part number JE232CM. Contact Jameco Electronics, 1355 Shoreway Road, Belmont CA 94002 or call them at (415) 592-8097. Note that you may be able to get this adapter at a discount if you order from the ads in the back of BYTE or other computer magazines.

Other adapters are available and may work as well, but the Jameco unit is readily available. I have had to dig in and modify other brands of RS232 adapters designed for the Commodore to make them work properly with Amateur digital communications, so Caveat Emptor (let the buyer beware!).

When you get the JE232CM, set all four DIPswitches on the JE232CM to the ON position.

Next, make a "null-modem" style RS-232 cable to go between the JE232CM and your TNC. A suitable schematic appears below:

```
   JE232CM                    TNC
   DB25P                      DB25P

     2    --------------------- 3
     3    --------------------- 2
     7    --------------------- 7
```

Note that this cable is not "polarized" -- you may plug either connector into the JE232CM and the remaining connector into the TNC.

This cable will depend on the use of software flow control (X-ON/X-OFF handshaking). Hardware flow control may also work with the cable wired appropriately, but I haven't tried it.

Features NOT supported

Hardware handshaking is not supported with this cabling, although the JE232CM and the TNC 2 are both capable of it. The simple terminal emulator program in this article does not support use of a printer, disk drive or cassette recorder for logging operation.

Software for VIC 20

The program below will convert between the Commodore character set and standard ASCII. The spaces are critical! Type this program in EXACTLY as shown or you will get messages like

```
?SYNTAX ERROR IN 370.
READY.
```

Of course, even though the program doesn't support disk drives or cassette recorders, you can save the program to disk or cassette for reloading at another time.

```
 90 POKE 36869,242
100 OPEN 5,2,3,CHR$(6)
110 DIM F%(255),T%(255)
200 FOR J=32 TO 64:T%(J)=J:NEXT
210 T%(13)=13:T%(20)=8:RV=18:CT=0
220 FOR J=65 TO 90:K=J+32:T%(J)=K:NEXT
230 FOR J=91 TO 95:T%(J)=J:NEXT
240 FOR J=193 TO 218:K=J-128:T%(J)=K:NEXT
250 T%(146)=16:T%(133)=3:T%(134)=19:T%(135)=17
260 FOR J=0 TO 255
270 K=T%(J)
280 IF K<>0 THEN F%(K)=J:F%(K+128)=J
290 NEXT
300 PRINT " "CHR$(147)
310 GET#5,A$
320 IF A$="" OR ST<>0 THEN 360
330 PRINT" "CHR$(157);CHR$(F%(ASC(A$)));
340 IF F%(ASC(A$))=34 THEN POKE 212,0
350 GOTO 310
360 PRINT CHR$(RV)" "CHR$(157);CHR$(146);:GET A$
370 IF A$<>"" THEN PRINT#5,CHR$(T%(ASC(A$)));
380 CT=CT+1
390 IF CT=8 THEN CT=0:RV=164-RV
410 GOTO 310
```

This program was taken from the Commodore 64 Programmer's Reference Guide, page 357. Two simple modifications were made. Line 90 assures that the computer will be in display mode 2 (UPPER and lower case) rather than the default mode 1 (UPPER case and GRAPHICS). Line 250 defines function keys f1 through f3 (CHR$(133) through CHR$(135)). Pressing f1 will issue a ctrl-C to the TNC, placing it in COMMAND mode from CONVERSE mode. Similarly, f3 sends a ctrl-S to the TNC (telling it to stop output to the computer to give you time to read the display) and f5 sends ctrl-Q (to resume output after a ctrl-S).

Software for C 64

The program is exactly the same as for the VIC-20, above, except for line 90 which changes to:

```
 90 POKE 53272,23
```

Setting up the TNC 2

Both of the above programs assume that the TNC will be operating at 300 baud, even parity, 7 bits, 1 stop bit. All you need to do on the TNC 2 is be sure that the rear-panel DIPswitch has switch 1 ON and switches 2 through 5 OFF (see TNC 2 System Manual, Chapter 2, Page 7, Table 2-4).

Operation

Turn off power to the TNC 2 and the computer (Commodore 64 or VIC 20), install the JE232 serial cartridge in the User Port (see Commodore 64 User's Guide, page 3 or Personal Computing on the VIC-20, page V). Attach the 3-wire RS-232 cable detailed above between the JE232CM and the TNC 2.

Turn on the computer (but NOT the TNC 2) and enter the above program. Double check to be sure you have everything correct. Then type the word RUN

and hit the RETURN key on the Commodore keyboard.

You should see the word RUN change to run (UPPER CASE to lower case). After a brief pause, the screen should clear and the cursor will blink in the upper left hand corner of the screen.

Now turn the TNC 2 on. After the LEDs cycle on the front of the TNC 2, the sign-on message will appear. If the sign-on contains a "|" (vertical bar) character, this character will not properly display -- it may not appear at all. Don't worry about it.

At the cmd: prompt, type

    disp m

and hit RETURN.

a list of monitor commands and their settings should appear on your screen. If this happens, you are in business! Proceed with checkout. When you are ready to operate, instead of pressing ctrl-C (which will do nothing) use the f1 function key to return to COMMAND mode from CONVERSE mode. Similarly, use the f3 and f5 function keys in place of ctrl-S and ctrl-Q for "flow" control.

For Further Information

There are plenty of programs that will work just fine with these computers and a TNC. Check with your local Commodore dealer, Commodore club, or a friend that knows something about computers.

Remember, when talking to a non-Ham about your intended application, that a TNC is the same as a dumb modem (dumb meaning "not auto-dial"). If you mention software for use with packet radio, you may get a blank stare!

Some packet manufacturers, such as MFJ and Kantronics, have special programs for use with Commodore computers and TNCs. You may want to check these out for your intended use. (At this writing, these programs are under $25.) Just be sure you can see a demo of the software BEFORE you buy, or get a money-back guarantee if you can't see a demo.

Finally, a commercial cartridge package that I have used successfully on packet with a Commodore 64 (but am not recommending as being the best available) is called TelStar 64, available from Eastern House, 3239 Linda Drive, Winston-Salem NC. 27106. You may call them at (919) 924-2889. This software supports TNC operation at 1200 baud on the user serial port. Please be aware that the documentation supplied with the program is hard to read, and often inaccurate. And again, I have not used this program with disk drives or printers (although the documentation claims it will support both).

TS-430S RECEIVER IF STRIP MODIFICATION FOR PROPER AGC ACTION

Eric Gustafson, N7CL

Most TS-430S transceivers will pump up the receiver AGC voltage when switching from transmit to receive. All TS-430 transceivers I have seen will do this to some degree, but the problem is greatly magnified when the modifications recommended by Kenwood to the value of C60 and C164 are done to allow the radio to turn around fast enough for AMTOR. This fault will only occur if there is a signal or noise in the receiver's passband when returning to the receive mode. While this is merely annoying in the SSB mode, it makes the TS-430S essentially useless for digital modes like AMTOR, PACKET, and high speed CW which require rapid turnaround of the transceiver's resources.

The fault is due to the method Kenwood chose to use to mute the receiver IF amplifier during transmit. The following modification will correct this fault in an otherwise excellent piece of radio equipment.

Schematic diagrams of the pertinent parts of the receiver IF circuits are shown as it appears both before and after the modification. Part numbers refer to those used in the TS-430S IF Unit (X48-1370-00) schematic found on page 31 of the Instruction Manual. Components added for the modification are referenced by value only.

The following is a step by step procedure to be used to perform the modification:

1) Solder a 1N4148 diode across a 47k resistor. Be sure to keep the diode leads as close to the resistor body as possible.

2) Unsolder the end of R33 which is not tied to the base of Q4 and lift this lead of R33 above the circuit board.

3) Install the assembly formed in step 1 in the hole vacated by R33. The anode end of the diode goes toward the circuit board.

4) Solder the free end of R33 to the remaining end of the diode/resistor assembly. The diode/resistor should be standing straight up on top of the circuit board.

5) Solder one end of C(t) to the junction of R33 and the resistor/diode assembly. Solder the other end of C(t) to the nearby ground jumper which is connected to the emitter of Q4. The value of C(t) should be 0.1 microfarad if the narrowest filter used in the radio is the 500 Hz CW filter. If the narrow CW filter (270 Hz) is used, the value if C(t) should be 0.3 microfarad.

6) Unsolder the ends of R35 & R36 which are connected to the collector of Q4. Lift these leads above the circuit board and solder them togehter above the board.

7) Under the circuit board, solder one end of a 15k resistor to the +8 volt line. (This is the trace which is connected to the unmodified end of R36.) Pass the remaining lead of the 15k resistor through one of the now vacant holes in the circuit trrace for Q4 collector and allow the lead to protrude above the circuit board. Solder this lead where it passes through the trace. The protruding lead will be used in a subsequent step.

8) Remove R34 from the circuit board. On the component side of the circuit board, place one lead of a 330 ohm resistor through the hole vacated by R34 at the source of Q1 and solder the lead under the board. This resistor should be allowed to stand straight up on the board. The remaining lead will be used in a subsequent step.

9) Unsolder the ground end of R48 and raise this lead above the circuit board.

10) In this step a bipolar switching transistor will be prepared for installation and installed on the component side of the circuit board. This transistor should be a 2N2222, 2N3904, or equivalent. It is best if a transistor with a plastic case is used. Ths transistor will be installed upside down with the emitter lead bent around to pass through the circuit board and the base and collector leads sticking straight up in the air above the board. Pass the emitter lead through the circuit board at the hole vacated by the grounded end of R34 and solder it under the board.

11) Solder the free end of the new 330 ohm R34 to the collector lead of the switch transistor.

12) Solder the switch transistor base lead to the protruding lead of the 15k resistor using a short length of insulated hook-up wire.

13) Using a short length of hook-up wire, solder the now free end of R48 to the collector of the new switch transistor.

For "almost QSK" operation such as needed for AMTOR, AEA recommends the following changes on the IF circuit board:

1) Remove D50.

2) Change C60 to a 4.7 microfarad value.

3) Change C164 to a .01 microfarad value.

To increase the effectiveness of TS-430S audio stages for Bell 103 tones, change the following components on the IF unit:

1) Change C96 to a .006 microfarad value.

2) Change C42 to a .001 microfarad value.

3) Change C153 to a 130 picofarad value.

Trim all excess leads. This completes the modification.

BEFORE MODIFICATION

AFTER MODIFICATION

BBS NEWS AND VIEWS
Tom Clark, W3IWI

Happy New Year's to all. If 1986 continues the frenetic packet pace of 1985, it's hard to forecast what the "hot topics" will be even a few months from now.

Reports from all corners of the U.S. indicate that the Packet Revolution continues unabated. The logs and user statistics derived from them show this most clearly. Just as an example I'll cite our experiences in the Balto/Wash area. Based on several different ways of measuring activity levels, I have been seeing the activity doubling every 4-8 months. Hardly a day goes by but what a new user or two shows up in the BBS logs; in November, a total of 163 users showed up and kept the BBS off the hook 32% of the time! This growth is seen dramatically in the sheer volume of logs kept by my BBS in the following table:

```
Log file      Size    0    50   100  150  200  250k
----------    ------  |----+----!----+----!----+
LOG-SEP.84 =  11008   |*
LOG-OCT.84 =  43136   |****
LOG-NOV.84 =  81920   |********
LOG-DEC.84 =  96512   |**********
LOG-JAN.85 = 114688   |***********
LOG-FEB.85 =  97024   |**********
LOG-MAR.85 = 126080   |*************
LOG-APR.85 = 100608   |***********
LOG-MAY.85 = 137088   |**************
LOG-JUN.85 = 115200   |************
LOG-JUL.85 =  92800   |*********
LOG-AUG.85 = 163584   |*****************
LOG-SEP.85 = 200320   |*********************
LOG-OCT.85 = 210816   |**********************
LOG-NOV.85 = 217984   |**********************
```

Discounting the July dip (when I was in KL7 for the month), a steady growth with each month al··ut 15k greater than the month before is indicated. My system here has already hit its first crisis -- WORLI's software cannot handle 5-digit message numbers and we crossed the 10,000 barrier on Dec. Second!

Comments from Hank, WORLI (representing New England) and Eric, WD6CMU (for the SFO Bay Area) indicate that other metropolitan areas are experiencing similar explosive growth patterns and resulting serious QRM problems.

US vs. THEM

This growth in turn has led to an "us vs. them" attitude developing between BBS users and people who want real-time QSO's. Attitudes which sound like "ban the BBSs to Siberia (or ban them to some obscure frequency, or turn them off except between midnite and 6 AM, or ....)!" are heard all too frequently these days from a very vocal faction of packeteers. It seems that, in the minds of some, BBSs are mischievous, unthinking, and dangerous golems.

But the BBSs only do what their users (or abusers) tell them to do. Many packeteers strongly support BBSs because they do (at least) 4 things very well:

(1) They facilitate non-real-time communications (i.e. electronic mail)
(2) They are a good way to disseminate bulletins
(3) They allow users to engage in multi-way discussions (i.e. electronic conferencing)

(4) They are the closest thing we have to a long-haul packet radio network (by relaying electronic mail automatically).

The words "non-real-time" are the key to BBS activities; they are the epitome of store-and-forward packet radio. So it strikes me that those who advocate banning BBSs are really (whether they realize it or not) attempting to ban all automatic non-real-time activities!

SET YOUR TIMERS

We all can help alleviate channel congestion by fine-tuning our timers. The manual keyboard users (whether they are talking to another user or to a BBS) generate very little channel traffic and should have high priority; BBSs should have lower priority, and users transferring files between themselves should have lowest priority. The DWAIT and RESPTIME parameters are the key to doing this. In this area, WB6RQN and I have been pushing the following recommended values:

USER TYPE	Time msec	TNC1 DWAIT	TNC2 DWAIT
Digipeaters	0	0	0
Keyboard users	160	4	16
BBS/UNIX nodes, etc.	320	8	32
User file transfer	480	12	48

Note that the DWAIT=0 for digipeaters occurs automatically, regardless of what DWAIT is set to! We haven't done much experimenting with RESPTIME on the TNC2's, but prudence would indicate that it should be at least 4.

Another way we can all minimize channel congestion: Don't do "BBS DXing"! Make certain your links are solid (especially before trying to dump a large file). The odds are that packets are going to be lost and merely contribute to channel clog along the path. Most BBSs now have the ability to restrict the maximum number of digipeaters in a user's path, and most of us have limits of 3-4 (and in some cases less). Connecting thru too many digipeaters wastes everybody's time and our valuable channel resources.

THE HIDDEN STATION PROBLEM

Finally on this topic of BBS vs. user "wars", one of the main problems we all face is that of the "hidden station". The BBS and the remote user have no way to know what other stations the intermediate nodes may be hearing. As a result, everybody steps on everybody else; the channel clogs; nothing gets thru. There are several potential solutions to this problem, all of which require work, new technology and new ideas:

a. Segregation by frequency. BBSs should not be on the frequencies that other users use. The problem with this concept is that the BBSs tend to be "magnets" and users monitor BBS frequencies. The BBSs still collide with each other since the hidden stations are still there.

b. Duplex digipeaters. In the LA Basin (or as I call it, Disneyland Valley), the 146.145/.745 repeater consists of back-to-back 202 modems. This then makes a "dumb digipeater" in the AX.25 sense, but one on which all users hear all activity. There are no hidden stations. Although key-up times are longer than for

normal digipeaters, data goes through immediately, full-duplex, so channel throughuput is quite high. This solution requires that you have a repeater frequency pair, which in turn requires coordination with your local repeater council.

c. Segregation by geography. Cellular radio, normal FM repeaters, etc. have found that localizing the coverage of a core station makes a lot of sense. On packet, each cell would define a Local Area Network (LAN). The size of the cell can be tailored (QRP stations, small antennas, etc.) to match the number of users that can be supported in the LAN. Frequencies can be re-used a couple of cells away. All users are "local" so there are no hidden stations.

It is no secret that, for some time, I have been advocating the "cellular packet" option c. (and casting aspersions about the parentage and/or mental age of those who advocate segregation by frequency). Under the cellular concept one (or more) local coverage BBSs, UNIX hosts, network gateways, etc. would be in each LAN "cell". They would have high-speed links to similar facilities in nearby cells thru a "backbone" high-speed network. USERS WOULD NOT ACCESS THIS BACKBONE DIRECTLY!!!!

ONWARDS TO REAL NETWORKING
Various packet software gurus are now hard at work on "real" networking code which will really make this backbone network fly. Until then, we have to make do with better and smarter digipeaters plus BBS message forwarding to link the cellular nodes together.

Jon, KE3Z (at ARRL HQ) has developed dual-port digipeater (DPD) code and hardware to run on the Xerox 820 to help get this concept going. I have begun work on adapting a TNC2 be a DPD. Lyle, WA7GXD now has the new TAPR NNC running in the lab and it should be an excellent engine for smart multi-port digipeaters. The software and digital hardware seem do-able.

Steve, K9NG, has developed 9600 baud modems to to make the backbone run faster. Skip, WB6YMH has the K9NG modems working with Midland 13-509 radios on a 40 mile path in the LA area on 220 MHz. Tom, K4GFG has been trying to identify suitable radios for higher speed use. But finding adequate stocks of currently available, suitable radios which can be easily modified to take 9600 baud (or higher) data seems to be the current sticking point. Once again, packet radio has problems on the radio side!

WORLI NEWS

Hank Oredson, WORLI has spent a lot of time recently trying to sort out the subtle handshaking differences between the TNC1 and TNC2. Howie, N2WX and I have been spurring him on, testing code, making suggestions, etc. In late November Hank sent a few of us his Release 10.4 code which seems to have solved most problems, and a recent discussion with Hank indicated that 10.6 was "solid" and should be in the mail before the end of the year. I have been using a dual-port RLI system with an antique "beta" TNC1 and a TNC2 for several months here. My impression is that the TNC2 operates much more reliably than the TNC1; it never "locks up" or "goes to sleep"! And the 10.4 code seems to have solved the residual problems of losing characters when the 820 is writing to the disk. Good show, Hank!

Hank continues to voice frustration over the problems of keeping the howling masses updated with new software releases. If Hank only knew how extensive the underground "copy the latest 'RLI disk for me" network really is! Hank also informed me that his HF gateway/BBS has now earned a packet WAC (if he can get cards) with the appearance of a Japanese station to give him Asia.

I was in San Francisco in early December and took the opportunity to meet with some of the Bay Area packeteers and to try their systems (everybody carries a TNC2 in their suitcase, right?). I logged onto W6CUS-1 BBS on 145.09 and was pleasantly surprised to find that they were running a 10 Meg hard disk on an RLI 820 system. Bill, N6FQR has made a prototype SASI interface which seems to work very well; I prodded Bill to write it up ASAP since a number of people would be very interested. Bill (and later Hank) told me that Tod, KO7O was working on a simple SASI interface that uses the parallel printer port on the 820.

Hank also informs me that his code has been successfully ported to the Ferguson Big Board, that KE7CZ has ported it onto a Kaypro 4 and that KI4XO is working on a version for the Kaypro 10 (with hard disk support).

IBM-PC EFFORTS

In last October's column, I commented that now was the time to begin working on new code to exploit multi-connect TNCs. I suggested that this code should exploit some of the more readily available off-the-shelf computers like the IBM-PC (and its clones). I received a number of "right-on" comments. It seems that the hassle of getting a Xerox 820, finding a keyboard, scrounging a keyboard and disk drives, making cables, putting it into a box, etc. requires too much an investment (of time) for a lot of folks. PC-clones costing under $1k are a much more attractive idea.

One set of PC BBS software was made available to the community by WA5SZL and the Raleigh/Durham NC group. This can be obtained by sending a diskette, mailer and return postage to:

Randy Ray, WA5SZL
9401 Taurus Court
Raleigh NC 27612

I obtained a copy of this and was disappointed to find that it did not support message forwarding as defined by the WORLI software; I understand that they plan to add this capability in subsequent releases. I also noted that the user interaction was more verbose than I would have liked, especially with the busy channels we see in the larger metropolitan areas. I was disappointed to find that only object code was being distributed.

Another exciting option is on the horizon. Like a bolt out of the blue in November an announcement appeared on Compuserve that Jeff Jacobsen, WA7MBL (in far-off Logan, Utah) had translated WORLI's Z80 assembler code into Turbo Pascal to run on the IBM-PC, and that he was looking for "beta" testers. Based on positive comments from AA4RE and W3VS who have been testing it in the San Jose area, I got added to Jeff's "beta" group and have his release 1.04 running as a second BBS (W3IWI-1) on my Zenith 150 with hard disk. I have only positive praise for this software. The bugs were minor and it looks like an 'RLI system to the user. Jeff still has to sort out his distribution

scheme and his interfaces with Hank so that development of both the 820 and PC versions can proceed in parallel, so PLEASE!!!!! don't bombard him with requests for disks!

SOFTWARE DISTRIBUTION

All this leads me to my final topic -- software distribution. A software developer like WORLI, WA7MBL or WA5SZL finds himself torn by a series of dilemmas:

a. Should I act as the distributor (and spend all my time making disks and answering questions) or should I devote my efforts to improving/augmenting the code and fixing inevitable bugs?

b. How can I recoup my out-of-pocket costs on this project? These costs may include subscription fees for commercial electronic mail networks, disks, mailers, special hardware/software needed for development and testing, etc.

c. How can I maintain some degree of configuration control? Should I distribute source code? I don't want to have to debug somebody else's mutilations, and yet I want to encourage others to add their ideas.

d. Can I come up with some way so that our hobby can derive tangible benefits from my efforts?

e. How can I minimize the personal hassles, midnight phone calls, etc.?

I faced a similar problem a few years back in AMSAT. I had written a program for satellite tracking in a somewhat obscure (North-Star) BASIC dialect. I wanted to see that it was translated, augmented and made available. With the help of Bob, N5AHD (and later Bob, N4HY) we set up the AMSAT Software Exchange (ASE). Bob and I solicited code translators and software developers. We also found people who were able to act as disk cloners. Bob initially handled distribution from his home, but we transferred that function to Martha at the AMSAT office. People wanting disks were asked to make a contribution to AMSAT, partially to defray the costs and partially to help fund building satellites.

My original copyright notice was placed on each piece of software, along with an addendum from the translator. This notice made the software available freely for amateur use. We knew that there would be copies made in the "field", but as often as not we received a contribution from the surreptitious copies (this is quite parallel to the late Andy Flugelman's "Freeware" experience with PC-TALK).

So this brings me to my modest proposal: TAPR should set up a PSE (Packet Software Exchange) along the lines of ASE. For software that is evolving (RLI seems to have a new release about every month) subscriptions could be arranged. We would need to have volunteers who could clone disks (unless we chose to use commercial duplication services when the volume warrants). I plan to propose this to TAPR's Board of Directors in February and would like feedback from you.

NEW PRODUCTS FROM TAPR

In addition to the NNC project, TAPR is working on several other fronts to enhance packet operation.

Tuning Indicator

We have received numerous requests over the past year and a half for an easy-to-build tuning indicator for use on HF and OSCAR packet work. In the June, 1984 PSR a tuning indicator was described that was based on information generated by the XR2211 PLL demodulator. This unit has been successfully employed by a number of Amateurs around the world. Still, it wasn't available as a kit, and the display was not considered optimum by many.

The last issue of PSR Quarterly contained an article by Dan Vester, K27CZ describing an improved tuning indicator for use with TNC 1 and TNC 2. This tuning indicator features a single moving dot display and easy calibration. It is suitable for HF as well as other weak signal work.

Dan has agreed to allow TAPR to produce this device as a kit.

A circuit board layout has been accomplished and the first boards should be in testing by the time you read this. If all goes well, we should be able to deliver the first tuning indicator kits at the TAPR Annual Meeting in February. Expected price is about $25.

There will be a general announcement of availability soon. Please do not send any orders to the office until that time.

LSC-10

The ALJ-1000, included free with every TNC 2 kit that TAPR paroduced, is not history. In its brief life, it gave joy to many packeteers. TAPR salutes the RMPRA for their efforts in this important area of packet development.

The ALJ-1000 was a useful accessory for constructing a TNC 2 kit. But we needed an accessory that would be equally useful for packet operation.

We believe we have found the accessory that any self-respecting packeteer will be proud to own.

At the TAPR Annual Meeting, Pete Eaton, WB9FLW, will unveil this useful packet station adjunct. Of all solid-state architecture, the LSC-10 uses the latest technology to bring you years of trouble-free operation.

The first production run is scheduled for delivery to TAPR in early January. Assuming field testing is satisfactory, the first units will be delivered at the TAPR Annual Meeting. The cost will be only $5.00 at the meeting.

TNC2 - C10 REPLACEMENT PART

For those of you looking for capacitors to replace C10 on the TNC2, you might want to check Radio Shack Part # 272-1027, 50 uf @ 35V, which fits the board like it was made for it. The price is $0.69.

DEPT. OF PROGNOSTICATION:
FUN WITH NUMBERS
Harold Price, NK6K

How many TNCs are there? (And how many will there be?)

Aside from being an answer to a trivia question, it is one of many items that must be considered by network planners.

The installed base, as of the end of October, 1985, by my estimate is: 8750.

This number is based on actual TAPR numbers (3700), and information from other vendors. The other vendor numbers are a blend of (1) their numbers (high), (2) their competitor's estimate (low), and (3) various other information sources. Results are checked for credibility against published packet census data from various locations. The number includes TNCs shipped overseas, but does not include TNCs manufactured offshore, nor does it include bare board TNC-1s manufactured and sold worldwide under TAPR's OEM agreements.

These numbers are within 15 percent, and are conservative. It isn't sporting to break out the list by vendor.

An estimate of the money spent on TNCs so far is $1,892,000.00 (One million, eight hundred ninety-two thousand dollars).

The Heath and Kantronics units have only been on available since April 1985, the TAPR TNC-2 since August 1985. A little more than half of all TNCs in existence have been sold in just the last 6 months.

Some major amateur digital milestones.

September 1978
Non-baudot digital transmissions made legal in Canada. Digital experimentation begins.

January 1979
VADCG group formed. This group produced the VADCG TNC, some are still in use today.

Summer 1979
Work begins in Ottawa and Mon treal. Total North American digital users: less than 30.

March 1980
ASCII data legalized in U.S. Canadian missionaries armed with VADCG TNCs and software cross the border.

December 1980
First U.S. digipeater goes on the air in San Francisco, it uses homebrew hardware and software based on the VADCG protocol (now called V1).

1981
First great packet diaspora begins. VADCG distributes PC boards. Homebrew systems are developed. Most areas standardize on 1200 baud bell 202 modems and VADCG compatible hardware. Locally maintained software versions in San Francisco, Washington DC, Vancouver, and elsewhere begin to diverge.

October 1982.
AMSAT and AMRAD host another in a series of meetings to solve the divergence problem by developing a protocol standard. Other major goals include the desire to support more than the 32, 64,

or 128 users allowed by then current V1 implementations. The AX.25 standard is born. Total North American digital users: no more than 200.

January 1983.
After several months of design and testing, TAPR produces 170 assembled and tested TNCs.

October 1983.
TAPR TNC kit (now called TNC-1) is beta tested by 19 users.

December 1983.
200 TNC-1 kits are shipped. In the mean time, more VADCG boards were assembled. GLB takes out first ad in QST for an assembled and tested unit. Total TNCs: about 650.

1984.
TAPR begins to ship TNC-1 in bulk. They ship an average of 120 TNCs/month for the next 15 months. AEA announces an assembled TAPR TNC-1 clone at the Dayton Hamvention. Packet hits the big time when Lyle, WA7GXD, wins the Dayton Technical Excellence Award, he accepts on behalf of packet radio and TAPR. AEA legitimizes packet by placing the first full page ads in the big ham magazines. At the end of 1984 there are more than 2500 TNCs.

1985.
Heath announces HD4040 TNC-1 clone kit. Begins shipping in April, sells out first 500 in three weeks. Kantronics announces "Packet Communicator". TAPR announces TNC-2. GLB announces PK1L. AEA announces PK-64. AEA, GLB, MFJ, and Pac-Comm announce TNC-2 clones. During the late Summer, most of the packet industry is "sold out", with demand far exceeding production.

Summary: 1982 200 TNCs.
1983 650 TNCs.
1984 2500 TNCs.
1985 10000 TNCs.(projected, 8750 through Oct 1985.)

Where will we be in 1986?

AEA has announced the PK-64. Targeted are all those C-64s setting in closets out in Ham-Land, it is the first low priced unit to include Packet, RTTY, and AMTOR in the same box. Industry-wide, expect at least another 5000 TNCs by the next ARRL networking conference in March, 1986. That's about 14,000 TNCs.

How far can we go?

The 1985 Callbook lists about 460,000 amateurs in North America. A study commissioned by the ARRL a few years ago found that half of the amateurs polled considered themselves "active". Perhaps a better indication for our purposes is the number of RTTY units sold during the big computer-RTTY boom of a few years ago. A discussion with two vendors at the Dallas hamvention early this year and with another vendor this week resulted in a guess of 50,000 units sold to 30,000 individuals.

I believe our growth will peak somewhere between this figure and the total number of 2 meter HTs. I can't get a good guess for that last figure, anyone want to take a stab?

I also believe that for Amateur Radio to survive in the long run, we'll need new blood. A large percentage of new hams will be interested in digital radio, many will be drawn to our hobby solely for its digital aspects. So in the end, who can say how far we'll go?

ENGINEERING A PACKET SWITCH
Phil Karn, KA9Q

Here are some specifics from a queueing theory textbook on how much buffer memory is likely to be needed in a packet switch. This is from the fundamental "M/M/1" queueing model, which makes the following assumptions:

1. There are many independent sources all generating packets at random intervals, but at a constant, predictable rate when averaged over a long time. (This is the first "M" in "M/M/1", which means "Markov arrivals".) The statistics of such a collection of sources is described by a Poisson distribution.

2. The time needed to process each packet (i.e., the time to transmit it on the output link) is also randomly distributed, but with an exponential distribution. Again, the size of each packet is random and independent, but there are predictable long term statistics. (This is the second "M", which means "Markov service times".)

3. There is one server (the "1"). This is the output link in our packet switch. If you have more than one such link, you treat the system as several independent queues because the links aren't interchangeable; you wouldn't want to send a packet on the wrong link because the right one is busy (although there is an interesting routing algorithm called the "hot potato" strategy that does exactly this.)

Now we define "p" as the average, long term utilization of the server (i.e., output link). "p" ranges between 0 and 1, e.g., p = 0 means that there is never any traffic, while p = 1 means that the link is always busy.

Knowing "p", you can predict the long term average length of the queue waiting for the output link, and knowing Es, the average time to transmit a packet (i.e., the average packet size divided by the link speed) you can predict the average delay a packet will encounter:

Average number of packets in system:
$$L = p/(1-p)$$
Average number of packets waiting in the queue:
$$Lq = p^2/(1-p)$$
Average time packet spends in system:
$$W = Es/(1-p)$$
Average time packet spends waiting in the queue:
$$Wq = p*Es/(1-p)$$

Note that as p approaches 1, all of these numbers approach infinity.

For our purposes we want L and W, rather than Lq and Wq, because we need to buffer the packet during transmission as well as when it is waiting its turn. Therefore, if you have an output link which you want to run at 95% capacity, you will need an AVERAGE of .95/(1-.95) = 19 buffers for that link. Of course, this is only an average; there is a 5% chance that at any given instant there will be no packets at all in the system (i.e., 5% of the time the link will be idle) and there is also a non-negligible probability that at a given instant there will be more than 19.

The exact formula for the probability that there will be more than N packets in the system is simply p^N, and this is never zero for non-zero values of p, no matter how large N is, although it

may be made very small. In our case it is $.95^{19}$ = .38, so if you only provided for the average of 19 packets there is a 38% probability that you'd have to drop a packet because of lack of memory. The more buffers you provide above the expected average, the lower the probability that you'll have to drop a packet, although you can never make it zero.

For example, if we wanted to engineer our system to run at 95% capacity and we wanted to reduce the probability of dropping a packet to below .001 (0.1%), then we would need log-base-.95(.001) = log(.001)/log(.95) = 134.7 buffers on hand even though we would only use 19 on average.

This model, although idealized, is a good start in analyzing queueing performance and buffer requirements. If anything, it is rather optimistic when compared with the real world. In other words, we will probably need even more memory than the model would predict because you're typically dealing with a smaller user population, and there are larger statistical fluctuations in a smaller group (the law of large numbers.) Also, packet sizes in real systems tend not to be randomly distributed, but rather bimodally distributed. Roughly half or more of your packets are of the maximum allowable size, which means they take longer to transmit than an exponential distribution would indicate. A large packet is also likely to be immediately followed by several more large packets, i.e., transmissions are very bursty. (This also says that the maximum allowable packet sizes in many networks are too small.)

The simple moral: don't scrimp on memory in a packet switch, especially if you expect your links to be heavily loaded. If you do, you run the risk of 1) dropping packets or 2) invoking user-to-network flow control when it isn't needed (and letting the link go idle when it could have been used), or 3) both.

REPLACEMENT PARTS PRICE LIST TNC-2

The following replacement parts price list is offered as a convenience. You can probably do better shopping elsewhere for your spare parts.

TAPR P/N	Description	Price Each
RESISTORS		
CFR1/4-XXX	Resistor, 1/4 watt, 5%	$ 0.25
CFR1/8-XXXX	Resistor, 1/8 watt, 1%	0.25
CPR2-XXX	Resistor, 2 watt, 5%	0.50
68WR-XXX	Trimpot	2.50
CAPACITORS		
MONO-223	Capacitor, 0.022 uF, COG	3.50
MONO-XXX	Capacitor (except MONO-223)	0.50
RADXXX-XXX	Capacitor, Radial Electrolytic	0.50
AXI25V-108	Capacitor, Axial, 1000 uF	0.75
TRA16V-105	Capacitor, Tantalum, 1uF	0.75
TCAP-600	Capacitor, Trimmer, 60 pF	1.50
INTEGRATED CIRCUITS		
MF-10	IC	4.00
LM324	IC	1.25
LM358	IC	1.25
LM556	IC	1.25
XR2206	IC	5.00
XR2211	IC	5.00
27C64	IC, "STATE MACHINE"	8.00
27C256	IC, w/latest software	12.00
6264LP-XX	IC	8.00
Z84C00	IC, CPU	8.00

TAPR P/N	Description	Price Each
Z84C40	IC, SIO/0	12.00
CD4040B	IC	1.25
74HC4066	IC	2.50
74HCT04	IC	2.50
74HC14	IC	2.50
74HC107	IC	2.50
74HC139	IC	2.50
74HCT374	IC	2.50
74HC393	IC	2.50

SOCKETS

DIPS-08	Socket, IC, 8-pin	0.25
DIPS-14	Socket, IC, 14-pin	0.35
DIPS-16	Socket, IC, 16-pin	0.40
DIPS-20	Socket, IC, 20-pin	0.50
DIPS-28	Socket, IC, 28-pin	0.75
DIPS-40	Socket, IC, 40-pin	1.00

CONNECTORS

HM-02	Header, Male, 2-pin	0.30
HM-03	Header, Male, 3-pin	0.45
HMW-05	Header, Male, 5-pin w/wall	0.75
DIPH-14	DIP Header, 14-pin	1.00
DIPH-16	DIP header, 16-pin	1.25
DIN5-180S	5-pin DIN Conn, Female, PC	3.00
DIN5-180P	5-pin DIN Conn, Male, Cable	3.75
DB25PCR-S	25-pin RS232 Conn, PC	4.50
PWR2.1-R	2.1 mm Rt Angle PC Power Conn	2.50
PWR2.1-CA	2.1 mm Power Cable w/conn	2.50
JMP-02	Jumper, Push On	0.50

DIODES

1NXXXX	Diode	0.50
SR535D	LED	1.00

TRANSISTORS and VOLTAGE REGULATOR

2N390X	Transistor	0.50
VN10KM	VFET Transistor	1.50
7805CT	Voltage Regulator	2.50

MISCELLANEOUS

MS440-3/8	Machine Screw, 4-40 x 3/8	0.25
MW440-LOCK	#4 Lockwasher	0.25
MN440	Machine Nut, 4-40	0.25
BD08-AV	DIPswitch 8-position, rt angle	4.00
PBDPDT	Switch, Push-On/Push-Off	4.00
PBK-GREY	Cap, Grey, Push Button	1.00
XTAL-049	Crystal, 4.9152 MHz	4.00
INDL-100	Inductor, 10 uH, molded, large	1.50
INDS-100	Inductor, 10 uH, molded, small	1.50
TNC2-FT1	Stick-On Rubber Foot	0.25
TNC2-ST1	Self Tapping Screw	0.25
SJWIRE	Shielded Jumper Wire	2.50
HS-CLIP-1	Clip-On Heat Sink	0.50

PACKED SEPARATELY

CN1/3-PT	Battery, Lithium	5.00
TNC2-CAB	Cabinet, Extruded Aluminum	20.00
TNC2-BEZ	Bezel, Cabinet	3.00
TNC2-FPx	Panel, Front	3.00
TNC2-RPx	Panel, Rear	3.00
TNC2PC-R2	PC Board, Bare	25.00
	System Manual	8.00
	Assembly Manual	5.00

Note that some parts (like the PC board) may be out of stock. Such items will not be stocked in the future.

All parts orders should include a minimum of $2 for shipping and handling. Thank you.

PACKET CHANNEL CONGESTION

Many of you have undoubtedly noticed that the normal packet channel used in your area has become clogged beyond useability. There are a couple of reasons for this as well as a few solutions.

The heart of the problem is channel loading versus channel capacity. Most packet operation occurs at a data rate of 1200 bps on VHF FM. Due to the access methods employed, it is unlikely that a single packet channel can handle more than about 800 bps on a continuous basis, perhaps even less.

The solution is obvious. Reduce the offered load to the channel and the clogging will magically disappear. Unfortunately, implementing such a fix is not easy. After all, you can't ask everyone to cease operations, or start issuing Monday-only stickers!

Phil Karn, KA9Q, has outlined a possible solution as a proposed revision to the AX25 Level two specificaiton. This was presented at the December ARRL Digital Committee meeting, and is being investigated for practicality of implementation.

In the meantime, the following suggestions are offered.

1) Keep packet bulletin board systems off the frequencies used for "long-haul" digipeating.

2) Turn DIGI OFF on BBS stations after moving their frequency of operation.

3) BBS stations should operate with a large value for FRACK (like 20), MAXFRAME set to 1 and PACLEN set to 80. Then, if it is impractical to implement steps 1 and 2, above, at least the BBS station will not "hog" the channel.

4) If operating on HF or other noisy paths, use MAXFRAME 1, PACLEN 40 and FFRACK 10. It has been widely noted that auto-forwarding BBS stations often tie up 14.103 with retry after retry. Remember, the baud rate is 1/4 that of a VHF channel, and noise implies more retries with a given PACLEN setting.

5) Many operations on 14.103 are off frequency. This has caused interference to propogation beacons on 14.100. It is easier for packet stations to move than beacons, so please respect the beacons!

6) There is nothing magic about 145.010 or 14.103. Move local ragchews or file dumps off the "calling" frequency. Some folks have brapped CQs on 14.105 for hours with no response, yet heard stations calling CQ on a totally jammed 14.103. Remember to use your tuning knob!

7) Turn OFF ALL BEACONS! There is usually no good reason to beacon, and this ties up channels to an embarrassing degree. If you hear a station sending frequent beacons, or "cute" ones (that clear your screen, for example), note that stations callsign and get in touch with the operator. Use the landline, use tact, and explain the limited resources of the shared channel with him.

8) Turn OFF CWID. Again, this was implemented when packet was rare; now it is rare to run into an Amateur who hasn't at least some idea of what packet is. Of course, if you have a wide-coverage digipeater you may want to leave the CWID on.

9) If you must use the long-path channel for a file dump, do it on times other than peak operation. And set FRACK 20, PACLEN 80 and **MAXFRAME** 2. This will allow other stations to use the channel as well.

Remember, packet has tremendous opportunities for resource sharing. At the same time, it is the responsibility of every packet operator to share the limited frequency resources in a fair manner with other stations.

Be courteous, observe these guidelines, and...

...HAVE FUN!

TNC64: A PACKET RADIO TERMINAL PROGRAM FOR THE C-64
George Baker, W5YR

The Texas Packet Radio Society is distributing a new packet radio terminal program written for the popular Commodore 64 computer.

TNC64 is a terminal control program developed specifically for packet radio operation. It is written in both C-64 machine language and in BASIC to gain the best features of both programming approaches. Machine language code for the receive character handling and capture buffer management provides the speed required for up to 3600 bit-per-second operation. BASIC for the remainder of the program allows for ready enhancement and addition of features without major rewrites.

TNC64 offers these key features:
o 50,000-character capture buffer viewed on the terminal display, printed or saved to disk; AUTO-SAVE BUFFER mode for automatic saving to disk of up to three successive buffers of 150,000 characters; buffer never fills and locks up the program - it simply wraps and starts over if not saved; function keys control all buffer operations including ON/OFF toggling

o text files uploadable from disk to TNC o single keystrokes place TNC in either Command or onverse mode, and halt/resume transmissions from the TNC to the terminal

o validated with TAPR TNC1 and TNC2, AEA PK-1, Heathkit HD 4040, and Kantronics Packet Communicator

o text format control logic minimizes broken words and extraneous line feeds in both lower and upper case modes

o default selection of TNC/terminal communication parameters to simplify starting up the station; specification of individual parameters for other applications if desired

o plain-English menus to facilitate selection of modes and features; additional screen prompts provided where helpful

o easily loaded and placed in operation; no programming ability required; natural and convenient to use

o minimal required configuration of C-64, 1541 disk drive and monochrome display; full printer support provided but a printer is not essential; color display can be used

o pre-programmed Beacon Text loadable to the TNC at the touch of a function key

The C-64 is close to becoming the default ham shack computer, with many programs readily available for a variety of amateur radio applications. Now, the packet operator can put his C-64 to work as an advanced terminal that offers many features to enhance packet radio operation. No longer must he make do with programs written for telephone modem operation which do not support the unique needs of packet.

The key feature of TNC64 is its large 50,000-character capture buffer. A buffer of this size allows one to monitor for hours on end without exceeding buffer capacity and, as with some programs, locking the program. Further support of the monitoring function is provided by the selectable AUTO-SAVE BUFFER feature in which the buffer is automatically saved to disk each time it becomes full. After three saves the capacity of a new diskette is nearly reached, so the program automatically resets to normal operation.

All data that the TNC passes to the terminal is captured in the buffer, including all characters originated at the keyboard and echoed to the screen for operator display. Thus a complete record is captured not only of all other channel activity but also of both sides of the station's own contacts.

The buffer contents may be viewed on the terminal display, printed, or saved (either manually or automatically) to a diskette. It has not been possible within memory limitations of the C-64 to support printer or disk operation concurrent with normal terminal operation. Thus, it is necessary to go "off line" to view, print or save the buffer. These operations are selected from a menu using function keys, and TNC/terminal communication is automatically suspended until the off-line buffer operation is complete. Then, it is resumed, and data received in the interim is released to the terminal.

Text files stored on diskette can be uploaded to the TNC for transmission. Unlike some other programs, the uploaded text is also displayed on the terminal screen so that the operator can monitor the progress of the upload operation.

Although the current version of TNC64 is unique in its packet capabilities, a new version now in beta test is aimed at providing even more convenience and effectiveness. Version 2.0 will feature up to 20 user-specified memory-resident text phrases that can be selected for transmission to the TNC from a menu by the touch of a key or two.

Such commonly used phrases as CONNECT commands to various stations, QSO phrases, multiple formats of beacon text, UNPROTO identifications, and the like need no longer be laboriously keyed in each time they are needed. On-line editing of Key Text, as this feature is called, allows new phrases to be entered while the program is running; these can be made permanent later if desired.

This new version is being tested extensively in daily packet operations in the Dallas Metroplex before release planned for early 1986. Packet operators wanting to gain the advantages of TNC64 now, however, need not wait until Version 2.0 is ready for distribution. Earlier versions of TNC64 will be upgraded to Version 2.0 for a small fee upon receipt of the original TPRS program diskette.

TNC64 has been shipped to packet stations through-
out the US, and many operators have commented on
the pleasure of using a program developed es-
pecially for their packet radio needs. Newer
operators, particularly, appreciate the 20 pages
of detailed documentation provided with TNC64.

Each TNC64 program is provided on a first-quality
diskette that is individually tested in actual on-
the-air packet contacts before being shipped. Each
diskette is supported by warranty.

TNC64 is copyrighted by the Texas Packet Radio
Society and is being distributed on a contribution
basis to the packet community as one means for
supporting the development and construction of a
high-speed backbone packet network - TEXNET.

Further information about TNC64 can be obtained
from

> Texas Packet Radio Society
> P. O. Box 831566
> Richardson, Texas 75083

AN INTRODUCTION TO NETWORKS

part 3
T.C. McDermott, N5EG
networks SIG, TPRS

Before we get far into the discussion of network
software requirements, I would like to make a
correction to one of the previous articles. We
have chosen AX.25 as the link-layer protocol in-
ternal to the network, with the exception that the
maximum data size is 512 bytes. Since the user of
TEXNET may generate a packet up to 256 bytes long,
and since the network will overlay it's level-3
protocol onto that packet, then internal to the
network packets larger than 256 bytes can exist.
An alternative to this is to fragment packets, and
it was decided that this was an unnecessary com-
plication at this time.

In a previous article we saw that algorithms that
assume slightly unreliable radio paths can be
chosen to minimize the degradation in throughput
suffered by packets (the HOP-TO-HOP algorithm, for
example).

We have also seen that all the hardware that is
really necessary to build a node includes 2
radios, 2 modems, and the node control processor
(plus minor items like: sites, towers, feedlines,
power, people to maintain hardware, money, time,
etc.).

What then has delayed the introduction of networks
to the amateur community? Simply, it the great
level of complication in the software that is
necessary to build a network. We will see that a
network requires two layers of protocol, not one
layer, as we use in the AX.25 link layer. What are
the two layers of information, and what purpose do
they serve?

Let's define some terms here, since we will use
them frequently in the following discussion.
    LAN : Local Area Network. This is the part of
the network that the users are connected to the
nodes through. usually on 2-meters, at 1200 BPS.
    IP : InterProcessor. The part of the network
that the nodes talk to each other on. usually on
220 Mhz., at 9600 BPS.

source_user : this is the Ham, using a TNC,
that generates the original information to be
transmitted. This Ham may have typed it in on a
CRT, or may oe sending a disk file, or it may be a
BBS sending a message. It is the point where ASCII
text gets translated to HDLC.
    entry_node : This is the entry point to the
network. When a HAM wants to use the network, this
HAM will CONNECT to this node, via the normal
method for the TNC to connect to anything.
    exit_node : This is the exit point from the
network. It is the place where the node is close
to the desired ultimate consumer of the informa-
tion. The information leaves the network, via
AX.25 at this point.
    dest_user : This is the destination user, the
consumer of the information that is being trans-
mitted. Just as source_user is connected to
entry_node, dest_user is connected to exit_node
via a AX.25 link-level connection.

When you use a TNC to connect to someone, the
AX.25 packet contains two key pieces of informa-
tion, the source, and the destination. There is
never any confusion here. Source and Destination
change whenever the direction of transmission
changes. Also, since this is a link-layer proto-
col, there is never any confusion at to where
destination might be located. If it is not within
range of the radio, then no connection is ever
established!

What is different about a network? For one thing,
the network doesn't really "know" where the desti-
nation is. It must ask the source_user where to
find dest_user (or it might look up dest_user in a
table, but this gets complicated since users tend
to move around a lot). So the network needs to
know where to find dest_user, and the answer is
supplied as "dest_user is located NEAR to a par-
ticular network node, called exit_node".

Now, it would be nice for dest_user to be able to
send information back to source_user, who happens
to be located at entry_node. Thus when setting up
a connection, we see that the network, prior to
the exchange of user data, must establish where
the users are located, so that it can send that
data to the right place.

Several divergent opinions of how to get that user
data to that place could now burden this discus-
sion. Suffice it to say that we have chosen a
particular method for TEXNET for what we perceive
to have a simple implementation method.

Our method is as follows:
    1. Entry_node receives a packet from source_user.
    2. The network uses the AX.25 fields to tell it
who source_user is, and then strips off the AX.25
header.
    3. It now adds a brand-new header, called TEXNET
IP3 (InterProcessor layer-3). This header contains
the length of the data part of the packet,
exit_node, dest_user, entry_node, and a one byte
control field, followed by the packet.
    4. The exit_node field is examined by a routine
called ROUTE, which figures out which is the next
node of the network that should get the message.
This next node will obviously be one that is
closer to exit_node than this node is.
    5. An AX.25 header is now built onto the front of
this big packet. As the source field in this
header, it will contain our node name. As the
destination, it will contain the name of the
further-along node (that is adjacent to us) that
was supplied by ROUTE.

The circuit supervisory commands allow the two endpoints of the network to exchange information about setting up or taking down a connection. When setting up a connection, the entry_node can tell the exit_node who source_user, entry_node, and dest_user are, and any digipeaters that exit_node may have to go through to get to dest_user. Exit_node will either be sucessful at setting up the connection to dest_user, or it will not. It sends back this result to entry_node, so that source_user can be notified.

One of the previous articles described the special hardware state-machine on each NCP card that will detect a long sequence of characters that obey HDLC coding rules and directly generate a reset pulse to the microprocessor. The "fire code" command is the way that the network makes sure that the request gets to the node that needs to be reset, and that the node PRIOR to the target node will transmit the special sequence. Each node has a different state ROM, and thus a different sequence to reset it. Each node "knows" the code for it's neighbors.

Congestion control can be a very complicated subject. Congestion control and flow control ARE NOT THE SAME THING. As an illustration of this point, assume that source_user is generating packets quickly, and injecting them into entry_node. Entry_node routes them down the network to exit_node. Exit_node, however is having a difficult time trying to get the packets to dest_user. They get through, but there are a lot of collisions, and the leave exit_node slowly. Eventually, there will be a network congestion problem. User data will start filling up all of the available RAM at the intermediate nodes. Finally, the packets will back up through the network until entry_node no longer has any buffers to take packets from source_user. So entry_node will use the AX.25 RNR packet to tell source_user to flow-off. TOO LATE, the network is already hopelessly congested.

What should have happened is that exit_node noticed that packets were coming in for dest_user faster than they were being delivered. When exit_node notices the problem, it sends a message to entry_node telling source_user to stop (entry_node uses the AX.25 RNR). Now the network is not congested, and the source_user is stopped. Exit_node must tell entry_node to restart when conditions will allow. So entry_node has TWO conditions to check for:
1. Flow-off the source_user if entry_node is low on buffers, or
2. Flow-off the source_user if exit_node cannot deliver packets.
Only when BOTH conditions have cleared can source_user inject more packets into the network.

There are many other methods of congestion control, and this is obviously a simple one with some defficiencies, but it is easy to implement.

One of the difficult points to bring up is lack of a layer 3 protocol between the user and the node. Since only the link-layer protocol is currently defined in AX.25, certain problems arise in the operation of the network.

When source_user first connects to the network, he must engage an interactive question-and-answer session so that the network knows to whom source_user wants to connect. A layer 3 protocol would provide this facility connection establishment.

6. This complete unit will now be transmitted at 9600 baud, using all of the rules of AX.25, to the next node.
7. The next node will strip off the AX.25 header after it receives the packet. It will examine exit_node to see if exit_node = our_node_name. If it does match, then this node is the exit_node, goto 8, else we are not the exit_node, and the packet must be furtner propogated: Goto 4 to continue the propogation of the packet down the network.
8. When exit_node is reached, the field dest_user in the IP3 header is examined, and the correct information found for that packet. The IP3 header is stripped off, and the correct layer 2 AX.25 information, with this node name being the source, and dest_user being the destination is formed. This packet is then sent using AX.25 methods to the destination user. Although this sounds a little complicated, its really not.

INSIDE the network, any packet starts with a layer-2 header specifing which are the immediately adjacent nodes that are exchanging the packet. Next in the packet is the IP3 header, which contain the endpoints of the connection. Finally is the actual data itself. Note that we can place several different IP3-DATA sets within a single layer-2 envelope, as long as the different packets are going the same dirrection. Thus we have MULTI-PLEXED packets from different users into a single layer-2 unit.

This method has some advantages. One is that HOP-TO-HOP acknowledge is explicitly a function of layer-2, since each node must acknowledge the receipt from the previous noae by the AX.25 method. Secondly, an interviening node need only examine the exit_node field to decide if it needs to process the packet, otherwise the IP3 portion remains UNMODIFIED by that node. Layer 2 source and destination will be changed since the node that received the packet becomes the new source, and ROUTE will now supply a destination node closer to the exit_node.

The complete layer-3 protocol includes a few more details, such as a software state diagram to describe the exact method that the network uses to do everything, such as set-up and to tear-down a circuit. It also contains some instructions for handling error cases. Tne layer-3 macnine does not worry about the reliability of the radio paths, that is the responsibility of layer-2. It does however check for inconsistent activities, and erroneous values in tne control field of the IP3.

What does the control field contain? It tells us whether the information that follows is user_data (which it is most of the time), or whether the information that follows is supervisory information, for use ONLY within the network. The possible supervisory commands include these:
  Circuit establishment request
  Circuit establishment acknowledgement
  Circuit disconnect
  Circuit disconnect acknowlegement/ or
    establishment failure
  Error - Network transmission failure
  Error - User transmission failure
  Traffic statistics request
  Traffic statistics response
  Congestion control - flow on
  Congestion control - flow off
  Special "fire code" sequence request
  Processor reset acknowledge

PUBLIC DIGITAL RADIO SERVICE PETITION

The following is a condensation of a very significant proposal before the FCC. The complete text is available on Compuserve Data Library Zero as FCC1.DOC and FCC2.DOC.

Before the
FEDERAL COMMUNICATIONS COMMISSION
Washington, D.C. 20554
In the matter of
Creation of a new radio class
and allocation of spectrum for
the owners of personal computers

TO: The members of the Commission

PROPOSAL FOR THE CREATION OF
THE PUBLIC DIGITAL RADIO SERVICE

FILED BY
Donald L. Stoner, W6TNS
October 20 1985

SUMMARY OF PETITION

This petition is to identify the need for a new class of radio service. This radio service is described in the petition as the PUBLIC DIGITAL RADIO SERVICE.

The petition shows that creation of the service and the allocation of spectrum is in the public interest, convenience and necessity.

Presently, computer-to-computer communication by the general public is confined to the telephone network. Millions of computer owners find that it is increasingly expensive to utilize this network to satisfy their communication needs.

Establishment of the PUBLIC DIGITAL RADIO SERVICE would permit the owners of personal computers to communicate by radio. Instead of a traditional channelized scheme, the petition describes a radio Local Area Network (LAN). The PUBLIC DIGITAL RADIO SERVICE permits an infinite number of local area radio networks to be interconnected into a national packet radio network.

The PUBLIC DIGITAL RADIO SERVICE would allow computer owners to exchange messages, bulletins, computer programs and other information by radio, and at no cost.

BACKGROUND OF PETITIONER

I have been a radio amateur (W6TNS) since 1954. A large part of my career has been devoted to the field of writing. For an extended period, I was the Novice and Technician editor of CQ Magazine. I have written hundreds of articles and authored several books on the subject of amateur radio and computer communications. I was also responsible for the idea which grew to become the OSCAR satellite, and I was able to make useful contributions to the program during its early stages. I have been an educator and taught at Chaffey College in Southern California.

While modem communications will continue to be popular, an alternate cost-free communication path should be available to the computer public.

The computer public is not interested in the radio aspects of communication other than as a means to an end. Thus there is no need or desire for voice communications as part of an equivalent radio modem. This precludes the "chit-cnat" type of operation which was destructive on the Citizens Band.

Sharing frequencies with voice communication (such as on CB) would be unacceptable. Interference, caused by frequency sharing, would garble the received data. Since the interference is transparent, the typical user will assume that data errors are caused by equipment faults. Thus, it is essential that the frequency allocation for the PUBLIC DIGITAL RADIO SERVICE not be shared with any other service.

Channelized plans inevitably lead to a further problem. If the service becomes popular, there will ultimately be a need for more channels. This is exactly the situation which occurred on the Citizens Band. The Commission is well aware of the problems which resulted from the disruption of adding additional CB channels.

The alternative to a channelized scheme is to send the data at high rates using packet technology. A single wideband channel can be thought of as a digital highway with addressed packets entering and leaving the route in a highly organized manner (see "What Is A Packet Radio Network?").

AN ALLOCATION OF SPECTRUM FOR THE PUBLIC DIGITAL RADIO SERVICE

A wideband digital channel can only be accomodated within the VHF band or higher frequencies. To keep the cost of equipment low, a band between 30 and 300 mHz is ideal. Some readers may feel that a service as described should be placed in the UHF or SHF range. This might be true if a suitable allocation within the 30-300 mHz band did not exist.

However, within this frequency range there is a band, 2 mHz in width, which is virtually unoccupied and therefore unused. I refer to the spectrum between 52 and 54 mHz. Radio amateurs are permitted to operate on frequencies between 50 and 54 mHz (the six meter band). For a number of reasons, this band is "underoccupied".

It is estimated that out of 400,000 radio amateurs in the United States, less than 1,000 are active on the six meter band.

Due to the potential for inteference with adjacent television channel 2 (54-60 mHz), virtually all six meter users operate between 50 and 52 mHz. For all practical purposes the radio spectrum between 52 and 54 mHz is wasted.

The Cause of Interference- Radio amateurs have not used the 52.0- 54.0 mHz portion of the six meter band due to the high risk of television interference. This interference problem occurs through no fault of the amateur or the transmitting equipment.

Eliminating Interference- It is the opinion and experience of the writer that no televison interference can occur from a radio modem operating in the 52.0- 54.0 mHz band if the following conditions are met:

1. The signal strength of the TV station being received exceeds 100 uvolts.
2. The effective radiated power of an adjacent radio modem does not exceed one watt.
3. The separation between the radio modem antenna and the television antenna exceeds 8 meters.
4. The radio modem antenna is vertically polarized with respect to the horizontally polarized TV receiving antenna.
5. All modulation and spurious products which fall outside the authorized bandwidth conform to the FCC 43 plus 10 log10 rule.

WHAT IS A RADIO MODEM?

The device to control the node (see previous section) functions similar to a ham radio "digipeater" but at a much higher speed. Since the the term "digipeater" has no significance to the general public, the node controller is refered to as a "radio modem".

What is it?- Technically speaking, the radio modem is a non-persistent, carrier sense, multiple access with collision avoidance device. In practice, the radio modem consists of a small box, whip antenna and coaxial cable. The unit contains a receiver and transmitter, in addition to an RS-232 computer interface.

In addition to acting as a transceiving device, the radio modem is also capable of repeating received packets on the basis of a stored algorithm. In other words, it will receive, store and retransmit messages along the addressees route. Note that it is capable of acting as a repeater even if it is not connected to a computing device.

Training- Upon activation, the radio modem executes a stored training sequence. When first installed, the radio announces its presence and digital address in the network. The radio modem transmits its position with respect to other units, determines the digital address of other nearby units and finally, adjusts its power output to the minimum required to maintain communications with the other nearby units. This power can vary from 1 milliwatt for densely populated areas to the 1 watt maxiumum in rural areas. It is essential that the radio modem transmit only sufficient energy to maintain contact with other nearby radio modems (nodes).

Training the radio modem for power output insures that a minimum signal level is radiated by the antenna. The purpose is to minimize the possibility of television interference. Some readers may point out that one watt is simply not enough power for rural areas. However, it is not the purpose of the PUBLIC DIGITAL RADIO NETWORK to duplicate the elaborate trunks of the public telephone network. There are bound to be areas which cannot pass messages. Under no circumstances should consideration be given to increased power output in these instances. If a high power mode is available, it will be abused.

IDENTIFICATION- Enactment of a PUBLIC DIGITAL RADIO SERVICE will not affect the licensing workload of the Commission. Services which are essentially self-regulating (such as the remote control of objects, garage door openers, etc.) do not require the use of call letters. Inherent in the addressability of the radio modem, is a built-in aid to compliance and enforcement. Each radio modem has its own unique identification code, that is, its packet address. This is both the serial number and digital address of the unit. This code

also identifies the manufacturer and the physical location of the radio modem. Violations of technical requirements can be easily be correlated by manufacturer. In other words, if a significant number of units are observed to be defective, the manufacturer can be immediately determined by serial number correlation.

POWER OUTPUT- A major contributing factor to the "CB problem" was the addition of power amplifiers to CB radios in an effort to increase the talk range.

Adding a power amplifier to a radio modem will produce no increase in performance. The unit will "retrain" to reduce its power output to maintain the nominal signal level at nearby radio modems. Thus, the power delivered to the antenna might be 50 milliwatts (as an example), with or without the power amplifier.

ANTENNA- To further increase transmitting range, high gain, directional antennas were connected to CB radios. If the same type of antenna were connected to a radio modem, it would result in a "negative improvement". There would be no increase in range, since the radio modem would retrain to produce the nominal signal strength at nearby nodes. More important, the radio modem connected to a directive antenna could miss messages arriving from directions other than the antenna principal gain lobe. By the same token, raising the elevation of the antenna would cause no noticable increase in communication range.

OFF FREQUENCY OPERATION- Illegal out-of-band operation caused sizable headaches for the Commission enforcement personel. This will never be the case with the PUBLIC DIGITAL RADIO SYSTEM however. There is only one "channel" or band. If, by some means, the frequency of a radio modem were lowered, the data would be destroyed by amateur radio transmissions. If it were raised, video information from TV channel 2 would do the same thing.

TECHNICAL SPECIFICATIONS

The "radio modem" (node controller) to be used in the PUBLIC DIGITAL RADIO SERVICE shall meet the following specifications:

FREQUENCY BAND- Equipment authorized to operate in the PUBLIC DIGITAL RADIO SERVICE shall be capable of receiving and transmitting data within the band from 52.0 to 53.999 mHz.

MODULATION- The data shall frequency modulate the carrier in a frequency shift keyed scheme. Under no circumstances will equipment authorized for use in the PUBLIC DIGITAL RADIO SERVICE have provision for voice modulation or detection.

MODULATION AND SPURIOUS PRODUCTS- The data rate (see Note 1), waveform and signal processing shall be such that all products which fall outside the authorized bandwidth be suppressed by 43 plus 10 log10 (mean output power, in watts) decibels.

POWER OUTPUT- The power delivered by the final amplifier stage into a 72 ohm load shall not exceed 1.0 watts. Further, the radio modem (node controller) shall have an initial powerup "training" mode. Upon powerup, the power output will be 1 milliwatt.

The power will increase during "training" in 3 db. steps until contact is established with nearby radio modems (node controllers). This value is

stored in memory and becomes the nominal power output for the radio modem.

ANTENNA- The antenna shall consist of a vertical radiator which does not exceed one-quarter wavelength. The antenna shall exhibit no gain or directional characteristics. The antenna shall be supplied with a nominal length of coaxial cable.

TRANSMITTER IDENTIFICATION- Each radio modem shall have an imbedded identification which is transmitted as part of its packet address. The address will be used to identify the manufacturer, the serial number and the routing code of the equipment.

PACKET CONSTRUCTION- The packet and destination address will be contained in the header. The header will be constructed to limit the number of destination addresses. This is done to specifically preclude the transmission of "junk mail".

REMUNERATION- Users of the PUBLIC DIGITAL RADIO SERVICE shall be specifically prohibited from receiving any form of remuneration or compensation, either in the form of funds, goods or services, for handling data on the PUBLIC DIGITAL RADIO SERVICE (see Note 2).

TYPE ACCEPTANCE- Type acceptance procedures, similar to those for Citizens Band equipment, will be required. This insures that commercially manufactured equipment used in the PUBLIC DIGITAL RADIO SERVICE meets the specified technical requirements for this service.

NOTE 1- No data rate is given in these proposed specifications. It should be left to industry to determine the data rate. Schemes, unknown to the writer or Commission, may permit higher rates within the authorized bandwidth than conventional theory would dictate.

NOTE 2- The purpose of this provision is to prevent the use of the PUBLIC DIGITAL RADIO SERVICE for the benefit of common carriers.

The restriction should not be construed to preclude the use of the PUBLIC DIGITAL RADIO SERVICE for business applications. For example, the radio modem would be extremely useful within buildings to avoid the need for local area network cabling. It is likely the signals of an office radio LAN would not connect to the external PUBLIC DIGITAL RADIO SERVICE.

The reader might envision that the service would be usurped by the business community. This is not likely, however, due to the self-regulating nature of the PUBLIC DIGITAL RADIO SERVICE. Businesses are used to the near instantaneous response of telephone data communications.

In comparison, the message response of a packet radio network is relatively slow. Only small businesses would find these delays tolerable. These are the same business which can least afford the increase in telephone rates.

There is an analogy in the use of the Citizens Band. Numerous channels are available and the equipment is quite inexpensive. Even so, this band is seldom used for business purposes. There are simply too many disadvantages for the business community.

CONCLUSION

In response to this petition, the Commission may point out that there has been no popular "groundswell" to create a computer radio service. Likewise, there was no public interest in the creation of a television service in the 30's. However, in 1932, the Commission recognized the significance of television and allocated two bands for development of this new technology.

By the same token, the Commission recognized the impact that FM radio broadcasting would have on sound reproduction. In 1941 they allocated an eight mHz band to bring high fidelity sounds to the public.

In either case, there was very little awareness that such technologies were possible when the allocations were made.

The creation of a PUBLIC DIGITAL RADIO SERVICE is another instance where the Commission could take the initiative and create a new service in keeping with current technology.

International Regulations- Since the allocation is above 50 mHz, it appears that no international treaties would be involved in making the proposed allocation. Rather, it is likely that other countries would develop a similar service for their citizens.

Amateur Radio Opposition- It is safe to assume there will be sizable opposition to this petition by amateurs. The writer has been a radio amateur for 30 years. During this period, no permanant allocation has been "taken away" from the amateur radio fraternity.

However, there can be no defense by amateurs of the inactivity on meters. A reallocation of the frequencies requested would benefit the majority at virtually no expense to the minority.

Amateur Radio Colaboration- The principal purpose of this petition is to obtain an allocation for a public computer communication band. The writer would not object if this goal could be achieved as part of the Radio Amateur Service. The computer public would accept an administrative fee in return for access to the radio spectrum. However, they would never accept any sort of "testing" to achieve this goal.

The writer would like to thank the Commission for the opportunity to submit this petition. Further, the writer appreciates the consideration this petition will receive by the members of the Commission.

Signed 20 October, 1985

Donald L. Stoner, W6TNS
6014 E. Mercer Way
Mercer Island, Wa. 98040
(206) 232-6968

TNC 2 Assembly Manual, Rev 2

The parts list calls for a single 6264L 8k RAM chip. There should be two.

The section dealing with installing the ICs (page ??) claims socket U24 should be empty. Wrong again! This is where the second 6264 is to be installed!

When source_user first connects to the network, he must engage an interactive question-and-answer session so that the network knows to whom source_user wants to connect. A layer 3 protocol would provide this facility connection establishment.

When exit_node connects to dest_user, the destination use thinks it is connected to the network, the dest_user does not know the name of source_user unless the network specifically tells dest_user PRIOR to delivering traffic to dest_user. Again, this facility would be provided by a layer 3 protocol in AX.25.

Although this doesn't seem too major a point, consider the operation of BBS's with a network. You connect through a network to a BBS. What does the BBS think your callsign is? The callsign of the network, of course. How does a BBS perform mail-forwarding through a network? Each BBS could "kludge" a method to use to the network, but a standard method that works REGARDLESS of who designed, built, and programmed the network would really be nice. This should be in the domain of the AX.25 layer 3 protocol.

One of the objectives in the protocol we designed for the layer-3 inside of the network (IP) was to allow the operation of a device known as a Network Bulletin Board (NBBS). Since the TEXNET IP3 header contains enough information, a NBBS can be a device that talks to the network AT 9600 baud directly on the IP link. It must emulate a NCP, but it has the information to do that. Ideally, it would be a multi-user NBBS, since the IP supports multiple users per NCP.

For those who have read this far, this network can be described as the type that presents a virtual-circuit interface to the host, but uses a datagram method inside the subnetwork. This makes the network similar to ARPANET, and not like SNA, or DECNET. (See Tannenbaum, Fig. 5-4)[1].

Part 6 of this series will describe one way to put together a software package to perform these functions. Currently, we have psuedo-coded 30 software modules for TEXNET. Certain rules and programming metholodogy can help to ease the construction difficulties with a complex problem.

----

[1] I forgot to credit the fundamental work of the subject in one of my prior articles. "Computer Networks" by Andrew S. Tannenbaum; Prentice-Hall, Inc.

POTPOURRI

Fm: Dave Gilmour WA1QPD [Compuserve ID 76703,705]
To: All
Subj: Modifying TNC 1 Software

Does anyone have experience adding small routines to the TAPR 3.2 (I think!) software? I just want to add a tiny password routine in machine code to allow my TNC to sit on a network with some security against unauthorized access. Is there any documentation on "systems" routines to do I/O that can be called from a little RAM-resident routine. (I plan to load the routine via the debug mode after a hard boot and then jump to it whenever I'm finished a session). Any hackers out there?

TNC-2 UPGRADE INFORMATION

In response to numerous inquiries, TAPR is pleased to announce the availability of various upgrades to present owners of TNC 2 Rev 1 and certain Rev 2 boards.

Due to special purchases and price concessions from vendors, TAPR was able to increase the standard memory complement on TNC 2 from 16k of EPROM and 8k of RAM to 32k of CMOS EPROM and 16k of RAM. In addition, CMOS SIO chips (U21) were provided after approximately 600 TNC 2s had been shipped.

Each CMOS option specified below will reduce current drain by as much as 60 mA. All three options will reduce your NMOS TNC 2 current drain from about 260 mA nominal to about 110 mA nominal.

CMOS SIO option: $12 postpaid.

CMOS 27C64 (STATE, 0000 or 2000 code): $7.50 post paid.

Reprogram your present EPROMs: $1.00 per EPROM. EPROM must include a postage-prepaid, self-addressed mailer (NOT an envelope) with EPROM(s) in anti-static foam.

Thank you for your interest.

PACKET ADAPTER

Following in the proud tradition of the ALJ-1000 TAPR is pleased to announce the LSC-10!

The LSC-10 represents the best in talent and technology that our hobby has to offer.

It has been brought to our attention that our previous product announcement on the World Famous ALJ-1000 was not very descriptive. To reconcile this situation with the LSC-10 we have extracted a detailed description of this new unit from its specification sheet:

"This is indeed the best value ever in   !
The exceptional quality of   is second only to   . This is because the material composition of   is high in   and low in   . The result is a   , a   , high strength compared to weight and superior   ."

TAPR does not have the capabilities to produce this fine unit to it's exacting specifications. We have contracted with an outside firm to construct and deliver finish units to TAPR. The LSC-10 IS NOT A KIT! It is fully assembled and will be available in quantity at the TAPR annual meeting!

See You There!

[NOTE: This Product Announcement was submitted past the deadline for this issue. Unfortunately due to transmission difficulties with our phone lines certain gaps occurred in the text. We apologize for this inconvience and will strive to do better in the future. Here is a note from the project manager: "Gwyn: Boy I'm really sorry, as hard as I tried I just can't seem to upload the text to DRNET without gaps. Sigh...I guess this will just have to do......<<GRIN>>.....Pedro." ed.]

PTT MOD FOR TNC-2: CORRECTIONS

Tom Clark, W3IWI

My modification to the Rev.1 TNC2's has now appeared in several places, including PSR and CTM. This note is to offer a caution, caveat emptor, and to eat a little crow!

In the last paragraph I indicate that you can reverse the sense of the LED from bright=rcv, dim=xmit by attaching the 1k resistor to pin 9 of U14 instead of pin 8. >>>>>> DON'T DO IT!!!!! <<<<<<

The problem stems from the fact that U14 is used as a part of the BBRAM logic and is powered from the battery when the power is off. If you reverse the sense, then you will be driving battery current thru the LED and will kill your battery in short order.

The mod as I wrote it up (dim on xmit) works fine and doesn't hurt the battery since when the power is off the U14 pin 9 output is low and no current flows thru the LED.

Hence, please delete the last sentence of the last paragraph of my modification notes and forget I ever said it!

NOTICE to TNC 2 Owners.

1. Remove R14 and R15 when using a Radio Shack Model 1-4 series computer (maybe everyone should consider removing them).

2. Change R98 from 4.7k to 100k if using a TNC 2 Rev 2 (Rev 1 boards do not have R98).

3. Most RS-232 handshaking protocols work differently than on TNC 2. Consider cutting the trace from J1 pin 20 and adding a jumper from J1 pin 4 to the junction of R20 and R22 for "normal" CTS/RTS flow control.

4. For HF operation:
   a. If you operate primarily on HF, change C45 to about 0.60 uF (add a 0.47 uF capacitor in parallel with the existing 0.15 uF capacitor). This will help prevent DCD "chatter" due to noise. When you operate at 1200 baud, you may notice some sluggishness in the DCD LED response, but it will be minor.

5. The period of the watchdog timer can be extended by increasing R83 to 3.9 Megohms. This will prevent premature transmitter shutdown when lengthy queues of frames are being transmitted.

AMATEUR PSYCHOLOGY – FOOD FOR THOUGHT

Phil R. Karn, KA9Q

Here's another thing to keep in mind while discussing the merits of different ways to build amateur packet radio networks.

The average amateur is very individualistic. It is much more difficult to get him to contribute to a joint effort resulting in a shared system than it is for him to buy another piece of hardware for his shack. AMSAT sees this all the time; the same

person who gladly forks over $2000 for an HF KW linear he hardly needs balks at buying a $100 share of a communications satellite that will likely do far more for his communications capability. The linear he can park in his shack, twiddle the knobs and show off to his friends. The closest he ever gets to the satellite is a picture in the Amsat Satellite Journal; it isn't truly "real" to him.

What this says to me is that a network design philosophy that puts the lion's share of the "smarts" out in the "network" (as opposed to the amateur's shack) is going to rub against the grain of most amateurs; they won't support it. Oh, they'll all say they want it alright, until you ask them to kick in extra money to help support it. Given how much money most hams have invested in RF hardware just to avoid paying the telephone company, I suspect you'll find it easier to sell him a more expensive box than it'll be to sell him a cheap box that requires a bigger membership fee to talk beyond his own back yard.

EDITOR'S COLUMN

This is being written in the midst of the Christmas holidays. New callsigns are crowding my packet video display. I want to wish a Happy and Successful New Year to all my packet friends and associates, both new and old. I look forward to meeting the faces behind all those new callsigns. 1986 will surely be a year of great change in the packet world; I hope that all will find it a year of progress as well.

I appreciate all the kind words you readers have sent along about PSR Quarterly. I am proud of each issue, but only the editor knows how much better it could have been if there had been just a few more minutes...Where would we be without deadlines! The real credit for the PSR goes to the writers. Contributions have been very plentiful and timely. If you would like to see your project or ideas in print, just send me an article. I also wish to acknowledge and thank my art and layout assistant, Brad Voss, WB8ZPE. He contributes greatly to each issue, and similarly to the monthly FADCA>BEACON newsletter that I edit.

If you enjoy the PSR Quarterly, but wish it was published more frequently, I would like you to consider **PACKET RADIO MAGAZINE**. This new publication (PRM for short) is an expansion and outgrowth of the FADCA>BEACON. PRM will be published monthly and contain technical and operational material similar to PSRQ. It will also have imbedded newsletters of various participating packet organizations throughout the nation. Have your club newsletter editor or secretary write for information about participating. Individuals may still subscribe by joining FADCA. My home address appears on the rear PSR cover.

TAPR membership applications and renewals should be sent to the Tucson address. The PSR editor address is only for matters dealing with the content of the publication. Thank you.

Mark Baker

Mark was one of the "original six" and a founder of TAPR. Although not a licensed radio Amateur, he has been a strong supporter of TAPR and its goals from the beginning.

Mark has served TAPR as a Director, and was a vigorous participant in the definition of the TAPR/DA protocol.

As Vice President of Modular Mining Systems, Mark has secured access to MMS' laboratory facilities for TAPR's use. This access has been a significant factor in the ability of TAPR to design, develop, test and refine packet radio equipment.

Mark brings insight and experience to the TAPR Board of Directors.

Steve Goode, K9NG

Steve has been involved with TAPR since the Beta testing of the TAPR TNC1. During this time he performed Bit Error Rate testing of the on board modem in attempts to improve its performance. He has served as Vice President of the Chicago Area Packet Radio Association since its founding. He assited in procuring and setting up the equipment for Chicago's wide area digipeater. Steve was also involved in the design of the modem section of the TNC2. He has also designed a 9600 bps packet modem which TAPR has made available to begin testing of higher speed packet links. Steve has also assisted several people who have had problems building the TAPR TNC1 and TNC2 kits.

Steve hopes to continue helping TAPR advance the state of the art for Amateur packet radio.

Scott Loftesness, W3VS

Scott, 38, has been licensed since 1963 (first holding WN6IWS, then WB6IWS, WA2UFB, and, since 1976, W3VS). He began his amateur radio career as a CW operator - never having much to say on phone! Now, his first love is packet radio, having been exposed to this virus at the First APRL Computer Networking Conference held at the National Bureau of Standards in Gaithersburg, MD. Since that time, he's owned a TAPR Beta Board and is now active with a TAPR TNC-2 hooked to his IBM-PC and active as the W3VS digipeater covering southern Santa Clara County, California.

In addition to amateur radio, Scott runs the Ham-Net Special Interest Group on the CompuServe Information Service. HamNet has been "on the air" since 1981 covering all aspects of amateur radio - but with particular emphasis on packet radio developments. Besides ham radio and computers, Scott enjoys flying, although he's been suffering from the very common "not enough time" disease of late!

Lyle Johnson, WA7GXD

Lyle was one of the founders of TAPR and has been actively involved with TAPR activities since the first day.

He designed the Alpha TNC hardware, and coordinated the hardware development of the Beta TNC and the TNC 1 kit. He designed the CPU and error-correcting memory portions of the Digital Communications Experiment now flying aboard UoSAT/OSCAR 11. More recently, he has been active in the design and development of the TAPR Network Node Controller (NNC).

Lyle has also been active in writing about packet radio. In addition to various magazine articles, he has made significant contributions to all editions of the TAPR TNC Manuals. He has been a steady contributor to PSR and PSR Quarterly.

Lyle serves on the ARRL Ad Hoc Committee on Digital Communications. In addition to being a TAPR Director, he has served TAPR as Executive Vice President (1982-83) and President (1983-present). In April, 1984, he was privileged to receive the first-ever Dayton Hamvention Award for Technical Excellence on behalf of TAPR for achievements in Amateur packet radio.

Eric Gustafson, N7CL

Eric has been involved in Amateur packet radio since 1983, and was an active participant in the TAPR Beta TNC project.

He is the person most responsible for the TNC 1 modem modifications that were incorporated in TNC 1 Rev 3. In addition, to assist in his early experiments on HF packet, he designed the tuning indicator featured in the June, 1984 PSR.

Eric is the primary designer of the four-port modem board being developed for the TAPR NNC.

Gwyn Reedy, W1BEL

Gwyn, 43, has been licensed since 1954 and has been active in packet radio since 1981, and a member of TAPR since 1982. He was one of the nineteen TAPR Beta TNC test site coordinators in 1982-3, and also performed Beta testing of the TNC 2. He is currently serving TAPR as editor of the PSR Quarterly.

He is founder (along with Ted Huf, K4NTA) and President of the Florida Amateur Digital Communications Association (FADCA) which is the second largest (to TAPR) packet club in the nation. He is editor of the monthly PACKET RADIO MAGAZINE (formerly the FADCA > BEACON). He is a frequent speaker about packet radio at amateur gatherings.

Mark Baker _____

Steve Goode, K9NG _____

Scott Loftesness, W3VS _____

Lyle Johnson, WA7GXD _____

Eric Gustafson, N7CL _____

Gwyn Reedy, W1BEL _____

Mark your choice of 5 candidates. Ballots with more than 5 candidates marked will be disqualified.

Mail this ballot to the TAPR office.
THIS BALLOT MUST REACH TAPR BEFORE FEBRUARY 8TH

TAPR   ATTN: BALLOT
P.O. Box 22888
Tucson, AZ 85734

MEMBERSHIP APPLICATION

Tucson Amateur Packet Radio Corporation
P. O. Box 22888, Tucson, AZ 85734

Name:_____

Call
Sign:_____ License Class:_____

Address:_____

City &
State:_____ Zip Code:_____

Home
Phone:_____ Work Phone:_____

If you wish not to have any of the above information published in a membership list, indicate the items you wish supressed here:

_____

I hereby apply for membership in TAPR. I enclose $12.00 dues for one year.

Signature:_____ Date:_____

The Tucson Amateur Packet Radio Corporation is a nonprofit scientific research and development corporation. The Corporation is licensed 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 acquired during and obtained from such research.

The officers of the Tucson Amateur Packet Radio Corporation are:

    Lyle Johnson ..... WA7GXD ... President
    Pete Eaton .... WB9FLW ... Executive VPM
    Pat Snyder .... WA0TTW ... Secretary
    Dan Morrison ..... KV7B ..... Treasurer

The Packet Status Register is the official publication of the Tucson Amateur Packet Radio Corporation. Explicit permission is granted to reproduce any material appearing herein, providing credit is geven to the author and TAPR.

TAPR membership and PSR subscription mailing address:

    Tucson Amateur Packet Radio Corp.
    P.O. Box 22888
    Tucson, AZ 85734
    (602) 746-1166

PSR editorial submission address:

    PSR Editor
    812 Childers Loop
    Brandon, FL 33511
    (813) 689-3355

PACKET STATUS REGISTER QUARTERLY

TUCSON AMATEUR PACKET RADIO CORPORATION
P. O. BOX 22888
TUCSON, AZ 85734

Check your address label for membership expiration date.