

Packet Status Register

Tucson Amateur Packet Radio Corporation

October 1987
Issue #29

Published by:
Tucson Amateur Packet Radio
PO Box 22888
Tucson, AZ 85734
Phone: (602) 746-1166

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

In This Issue...

- WA7GXD's President's Corner
- Beginner's Corner: State Machines, Part Two
- Australian Rules Complicate Network Development
- TAPR Board of Directors Nominations Open
- TAPR Software/ROM Distribution Procedure
- AMNET: An Amateur Packet Switch Network
- N6NKF Recommends Changes to AX.25
- TAPR PSK Modem Update from WA7GXD
- NET/ROM Version 1.3 Update Information
- AA4RE's "in the Mailbox" Column on Packet Radio Bulletin Board Systems
- Digital Signal Processing Status Report from N4HY
- Letters to the Editor

President's Corner

by Lyle Johnson, WA7GXD

Wow!

The July issue of PSR, the first one edited by Scott Loftesness, W3VS, came the other day. All I can say is that it sure looked good! This issue should bring us current with a quarterly publishing schedule.

Now, if enough of you submit material, that may be increased to every two months. That all depends on you...

TAPR Board

Two major items need to be brought to your attention.

1) Gwyn Reedy, W1BEL, resigned from the TAPR Board of Directors in August. Gwyn served us on the Board for over a year and a half, and his inputs and insight will be sorely missed. This was all explained in the letter to members dated 01 September 1987 and included in the July issue of PSR.

The TAPR Articles of Incorporation allows the Board to fill vacancies by electing someone to fill the unexpired portion of that term.

The TAPR Board is in continuous session via electronic mail, and during the month of September Dr. Robert McGwier, N4HY, was nominated and elected to serve out Gwyn's term. Effective September 29th, 1987, Bob has been serving you in this capacity.

Bob is active in AMSAT and packet radio. He wrote the QUIKTRAK software that is distributed by the AMSAT Software Exchange. He is working closely with Tom Clark, W3IWI, in the DSP project. In short, Bob is a do-er, and we welcome him, along with his energy and his insights, to the TAPR Board of Directors.

2) It is time for nominations to the Board once again. Please see the article on "TAPR Board Nominations" for details.

Redondo Beach

Harold Price, NK6K, and Wally Linstruth, WA6JPR, along with the TRW Radio Club, did a super job at coordinating the Sixth ARRL Computer Conference on Saturday, 29 August. The Proceedings of that conference, available from the ARRL for \$10, is the best yet! (I can say that this year because I managed to sneak away with my wife for our first vacation together in 11 years, and writing a paper took a back seat to the trip, so I didn't write anything for this year's conference.)

If you don't want to spend a paltry 10 dollars, get your club to buy a copy for its library and then check it out. It is worth reading. There are papers on networking, modems, DSP, hardware, software — if it's packet radio, and it's current, it's mentioned!

Continued on page 2

Digital Signal Processing

The DSP project, headed by Tom Clark, W3IWI, and assisted by Bob McGwier, N4HY, gained a lot of momentum with several folks signing up and plinking down their \$525 for the special purchase Delanco-Spry board. There may be more on all this from one of them, and I don't want to steal their thunder, so look elsewhere in the PSR for a possible update on this project.

While on this subject, I would like to be sure that no one misunderstands what I wrote in the last PSR in the Beginner's Corner. The hardware device I was describing is not a cast-in-concrete design; it was only an opinion I had of what shape a DSP hardware project MAY take, not necessarily what it WILL take. And I suspect the for-real project will be pretty different. Take heart! I share the view with some others that DSP is the next big step, it's hot and it's going to be big. Stay tuned! Or, better yet, contact Tom or Bob and get involved!

PSK Modem

The PSK modem project is in high gear now. The first 200 kits shipped, and another 200, with significantly better documentation, are ready now. The price is \$100 plus shipping and handling (\$10 in the US and Canada, more elsewhere).

These modems work exceptionally well on FUJ/OSCAR 12, and are being used for earthbound applications as well. In fact, there is a 20 meter 1200 bps PSK BBS running in Japan now, along with at least one HF digipeater (sigh, I really find HF digipeaters of any kind repulsive, but...). PSK should work well for weak signal work, and some modems are now being modified to run at 300 bps. Maybe there will be a report in the next PSR on the mods.

Order yours today. We will likely not produce more than the 200 on the shelf now, and may try to locate a commercial firm to take this one over. If so, the price is likely to be higher...

At least one TNC 2 OEM (Fuji Digital Systems of Japan) is making a TNC with the 1200 bps PSK modem built in. The modem is selected by software command.

4X Report

While on the afore-mentioned vacation, I spent some time with some old friends (and some new ones) in Israel. I had the privilege of addressing the Israel Amateur Radio Club in Haifa, Israel, in late July. There was a good group there and we touched on many points of packet theory and operation.

They are building a network covering from the Galilee in the north to Tel Aviv and Jerusalem, with hopes of eventually reaching Eilat in the extreme south. The Israel Amateur Radio Club is helping fund much of the equipment, including PC clones for two BBSes. In addition, they appear to have secured the cooperation of a radio equipment manufacturer to donate some radios as well. This is a high-energy group which includes some bright minds. Expect to hear from 4X land!

Ontario

I had the privilege of speaking at the 3rd Packet Symposium in Barrie, Ontario, in mid-September. I thoroughly enjoyed the opportunity to address the group and met several active packeteers. The folks in Ontario appear to be well organized, and their network growth well managed. Again, a group with a bright future.

PACSAT

The U.S. Department of Energy has awarded a \$350,000 grant to VITA for PACSAT! And Pete Hoover has added \$175,000 in challenge grants to that figure. With just over 1/2 million dollars, PACSAT may yet be born!

High Speed Modems

TAPR has received a pair of GLB NET/LINK 220 radios for Beta testing. Eric, N7CL, and myself hope to get these on the air and shipping data around in a test mode in the next few weeks. They run direct FSK, are crystallized up for 220 Mhz, and run 19.2 kbps. Power output is in the 1 to 5 watt range.

Software

The TAPR office is now providing a number of software programs for packet radio. See the blurb elsewhere in this PSR. You may be in for a big surprise!

Anniversaries

In late November, after this PSR and before the next one, TAPR will be six (!) years old.

Annual Meeting

Start planing now to attend. The big weekend is that of February 20th, 1988. We normally meet the second weekend in February, but were pre-empted for next year by St. Valentine. Unfortunately, the Theatre at the Airport Embassy Suites Hotel is booked for the entire month of February, so we will have to find an alternate location this year. Details will appear in the next PSR. Until the January issue, have a happy holiday season, a prosperous New Year, and may your retries be few.

73, Lyle WA7GXD

Letter to the Editor

He's Hot About 4.0!

Hi Scott, Congrats on becoming editor for the psr. I have been a member for a long time, but this time around I feel that I won't renew my membership to TAPR.

Why? I am a PROUD owner of a TNC-1 that I have had for many years. I joined TAPR back when I bought the thing and now TAPR has abandoned me. I have written many letters to TAPR and to Lyle himself without an answer, let alone a solution. I have heard for many years about Version 4 Level 2 software for the TNC-1 and how it will be ready "any day now"!!! I keep seeing where improvement have been made for other tnc's and thank God that WA8DED has come forth with a level 2 software for us or we still be in the dark ages as far as protocol goes. I guess the best way to show my disatisfaction with TAPR is to withhold the bucks. Like I said I have written and called with no reply.

Thanks for letting air my griefs and sorry that you, being new had to be the one to get it, but no one else seemed to care. Please answer me as I probably won't get another psr as my dues run out this month.

Thanks again and good Luck

Joe Hahn, WD8NBA @ W8CQK

Beginner's Corner - State Machines, Part Two

by Lyle Johnson, WA7GXD

As you may recall from Part One (waaaaaaay back in the April, 1986 issue of PSR), a state machine is used in the TNC 2 to recover clock and data information from our NRZI-encoded packets and present this information to the SIO chip (which doesn't understand NRZI).

We discussed synchronous and asynchronous logic, and the concept of state.

A state machine is a combination of logic elements (like AND gates, OR gates, and flip-flops) in such a way as to produce an output based on (1) current input information and (2) the previous state (or previous set of inputs).

State diagrams are usually used for discussions such as these. State diagrams are basically little (or big!) bubbles with numbers inside and arrows pointing every which way. The more complex the state machine, the more bubbles and arrows.

The idea here is to give you a better feeling for what a state machine is, not make you either (a) confused or (b) an engineer. (I am not sure that there is a difference between the two choices, but some folks might, so I made the distinction...)

So, let's consider something a little less complicated than trying to derive clock and data from an NRZI packet signal.

Instead, let's look at a simplified method of extracting a clock signal only from the NRZI data stream. Note that the method discussed below isn't necessarily the one used in the TNC 2's state machine.

The requirements we will place on our system are the following:

- 1) The applied clock to our state machine is 8 times the known data rate of the incoming signal.
- 2) The clock output from our state machine should occur in the middle of the data bit time.
- 3) We will only allow our "clock

divider" (actually, it is a digital phase locked loop, or DPLL) to adjust itself by a factor of one clock-time at any clock pulse.

This differs from many DPLLs in that some allow us to step more than a single clock-time if the incoming data is more than a little bit off center, and we are using a times 8 clock rather than the more common times 16 or times 32.

If our received data is known to be coming at 1200 bits per second (bps), then our state machine clock should be:

$$8 \times 1200 = 9600 \text{ Hz.}$$

We will call our state machine clock SCLOCK to keep it straight in our minds from the clock we are trying to derive.

Our input to the state machine is:

NRZI data (abbreviated INDATA)

Our output will be a positive-going clock signal occurring at the 5th clock from an INDATA transition. This signal will be called:

Data Clock (abbreviated DCLOCK)

This is all fine and dandy, but we need to keep track of how many SCLOCK pulses have occurred since an transition of INDATA. Our SCLOCK rate is 8 times our DCLOCK output, so we need a divide-by-eight counter inside of our state machine as well. This means we need three (3) counter outputs as well ($2^3 = 8$). We will all these counters:

Least Significant Bit (COUNT0)
Second Bit (COUNT1)
Most Significant Bit (COUNT2)

These should be all the inputs and outputs we need. To review, we have:

- 1 state machine clock (SCLOCK)
- 1 independent input (INDATA)
- 1 desired output (SCLOCK)
- 3 intermediate (feedback) outputs (COUNT0-2)

To implement this state machine, then we will need a four-bit latch that will look all all the inputs (INDATA and COUNT0-2), some logic that goes between the input and output sides of

the latch (we will use a programmable read-only memory, or PROM) and a source of logic-level pulses running at 9600 Hz to strobe our latch.

Since we have a total of four inputs to our latch, we have $2^4 = 16$ possible states that we can be in.

Next, we need to look at our algorithm. The word "algorithm" is just shop-talk for a "method to solve the problem." Remember the abbreviations we are using for our signals? The computerese word for this is "mnemonic." It's pronounced new-mah-nick, and is used to influence your enemies and impress your friends. It just means "abbreviation."

Still with us? If not, review the material above until you think you have the general idea so far. Now, let's consider the ideal relationship between our input INDATA and our output DCLOCK. Recall that NRZI data, like we use in packet, has a change from 0 to 1, or 1 to 0, every time we want to send a value of 0, and no change (that is, it remains at either a 1 or a 0) when we want to send a value of 1. And, by a technique called bit-stuffing, we ensure that we send a 0 at least every fifth bit-time. Thus, we are assured of an INDATA transition at least every five bit-times.

Now, we want to synchronize our SCLOCK with the edges of INDATA, then generate our DCLOCK so that it goes from 0 to 1 at the 4th occurrence of SCLOCK after a transition of INDATA. If there is no transition of INDATA, we will generate our next DCLOCK pulse eight "ticks" of SCLOCK since the last DCLOCK pulse.

We want our DCLOCK to be a square-wave, so we will return it from the 1 level to the 0 level on the 8th tick of SCLOCK.

We can check to see if we are in synchronization with INDATA by looking for a transition between the 8th and the 1st ticks of SCLOCK. If so, we are in good shape.

If we see no transitions of INDATA during the entire string of 8 SCLOCKS, we will assume that a value of 1 is being sent by the distant station, hence there is no edge for us to check and we will not modify anything.

If we see a transition of INDATA between the 1st and the 4th SCLOCK

tick, we will drop back a count to shift the edge closer to occurring before the 1st tick.

If we see a transition of INDATA between the 4th and the 8th SCLOCK tick, we will skip a count to shift the edge closer to occurring after the 8th tick.

The above rules may seem a little confusing, but if you think about them you will see that, if we can obey them exactly, we will achieve our desired result of having a DCLOCK positive edge near the center of a bit time period of INDATA. (NOTE: for those of you who are astute, yes, I am cheating and calling the INDATA edge the beginning of a bit time, when, from the sending station's point of view, it is the center of a bit cell. The result is the same, though, for our purposes, and using an INDATA edge to define edges of bit times, rather than centers, is a little less confusing for most folks.)

Now, since we have to remember the previous level of INDATA to determine if a change occurred since the last tick of SCLOCK, we need to name it (to add/subtract from the confusion). We will call this "PREVIOUS INDATA" (PREDAT).

These rules are coded into the table below. The left side of the table tells the story to the instant of the SCLOCK tick, the right side result immediately after the tick.

Note that there are 32 entries in the table. 16 cover the possible states that our state machine can be in (2 values of PREDAT times 8 values of COUNT0-2), and these 16 are doubled since that is the possible levels the incoming INDATA signal can have.

Note further that DCLOCK is simply the value of COUNT2. It goes from 0 to 1 at the 4th tick of SCLOCK and drops back from 1 to 0 at the 8th tick. Convenient, huh?

Letter to the Editor

Enjoyed the July Issue

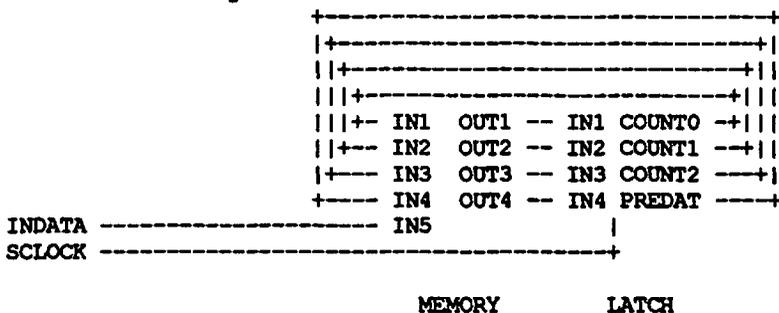
Scott: Congratulations on a good issue. Please feel free to use any part of the "Indiana Packet NTS Newsletter" in future issues. And let me know if I can be of any other assistance.

73, Jay WB9MDS

State Tables for DCLOCK State Machine

		DCLOCK			DCLOCK			
INDATA	PREDAT	COUNT0	COUNT1	COUNT2	COUNT0	COUNT1	COUNT2	PREDAT
0	0	0	0	0	1	0	0	0
1	0	0	0	0	1	0	0	1
0	1	0	0	0	1	0	0	0
1	1	0	0	0	1	0	0	1
0	0	1	0	0	0	1	0	0
1	0	1	0	0	1	1	0	1
0	1	1	0	0	1	1	0	0
1	1	1	0	0	0	1	0	1
0	0	0	1	0	1	1	0	0
1	0	0	1	0	0	0	1	1
0	1	0	1	0	0	0	1	0
1	1	0	1	0	1	1	0	1
0	0	1	1	0	0	0	1	0
1	0	1	1	0	1	0	1	1
0	1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1	1
0	0	0	0	1	1	0	1	0
1	0	0	0	1	0	0	1	1
0	1	0	0	1	0	0	1	0
1	1	0	0	1	1	0	1	1
0	0	1	0	1	0	1	1	0
1	0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	0
1	1	1	0	1	1	1	1	1
0	0	1	1	1	0	0	0	0
1	0	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	0
1	1	1	1	1	0	0	0	1

Our memory needs 5 address lines (there are 32 possible conditions in our table) and four output lines. We can use a portion of a cheap EPROM, like the 2716 or 2764 (which we did in the TNC 2 state machine), and wire up our latches like the drawing below.



The pattern of 1s and 0s given in the table are entered directly into the memory chip and permanently stored there.

Now, whenever we apply a clock (SCLOCK) that is eight times the rate of incoming data (INDATA), we will generate an output (DCLOCK, same as COUNT2) that occurs in the middle of a bit time of the incoming data stream. Now that we have recovered the clock signal from our NRZI data, what do you suppose it would take to also recover the data from the NRZI data stream? All we need is another output from our memory!

That's right, we don't even need to latch this output back into our set of inputs. Let's call the new output Out DATA (OTDATA).

I won't fill this PSR with the resulting table, but you might want to try to work it out. If you remind me, I will make the table for the next PSR! (Just what you wanted, right? HOMEWORK over the holidays!)

The additional rules will be:

If, at tick 1, the level of INDATA is the same as PREDAT, OTDAT will be a 1.

If, at tick 1, the level of INDATA is different than PREDAT, OTDAT will be a 0.

Not so tough, eh? Why not write the values in the table above just after the PREDAT column on the right side of the table and compare it to the answer in the next PSR!

Enjoy!

A Note to Network Implementors from Down Under

by Barry White, VK2AAB

Australian Amateur Packet Radio Association

We are writing in regard to your Network Node Controller project. We wish to inform you of our regulations in regard to the identification of packets.

The relevant paragraph reads as follows:

"(3) Each "packet" shall contain the originating station's identification, that of the destination station and the station transmitting (if different from the originating station)."

We would appreciate your informing the group responsible for the design of the software for the NNC as we would like to be able to consider the use of the TAPR NNC when available. The format of packets when viewed by a monitoring station should be as follows: (VK2RPH is in Sydney, VK2RPN is in Newcastle)

As seen in Sydney:
VK2AAB>VK2CZZ,VK2RPH Fol-
lowed by the information

As seen in Newcastle:
VK2AAB>VK2CZZ,VK2RPN Fol-
lowed by the information

What is required is similar to digitizing except only the callsign of the

transmitting repeater node is required, not all those along the way.

Within the network itself we believe the same situation may apply, ie VK2RPH>VK2RPN then an indication of the users callsigns, in my example VK2AAB and VK2CZZ. The only way that I can see to achieve this is for the callsigns of the users to be placed in the information field and then stripped off by the last network node controller in much the same way as NET/ROM removes the network information from the information fields.

We will probably approach the Dept of Communications with a view to removing the users identification from the network level 3 packets.

Both Software 2000's NET/ROM and PacComm's DR200s are illegal [in Australia - Ed.] because of their identification methods. We suspect that both these systems are illegal in the UK and New Zealand and I am surprised that they have not been questioned in the US. After all, there is no way that anyone can know who is sending packets to the originator of that connection. Also, at the requested station's end of the network, the changing of the repeater's callsign to that of the originator with the -15 SSID must be illegal anywhere.

I realise that the software writers are probably well down the track of development at this time but if it would be possible to either incorporate our regulations into the design or produce a different version to suit it may be considered more suitable in many other countries than Australia.

More About Australia

by Phil Karn, KA9Q

A recent item on the local WA2SNA-1 packet BBS reports that the NET/ROM network in Australia was ordered to shut down by the DoC (the Australian FCC) because it did not meet that agency's requirements for amateur packet radio identification. Recently I spoke by phone with John Tanner VK2ZXQ, an active packeteer living in Sydney, who confirmed this unfortunate development.

I began digging through my collection of Digital Committee mailings for some background. I found "Review of Amateur Radio Service Packet Communi-

cations — Policy Paper from the Wireless Institute of Australia" issue 3.1, dated 20 October 1986. At the end is a summary of the rules issued by the DoC on 30 September 1986. Although they are certainly less liberal than our own, I found them to be clear and concise.

John mentioned that several other countries (mostly European) are looking closely at the Australian model as they draw up rules for store-and-forward packet operation in their own amateur services. The Australian rules therefore deserve a close look because they imply that certain networking protocols and techniques may not be acceptable to a significant fraction of the world's amateur population.

What follows are relevant excerpts from the rules and my own personal interpretations of how they should influence our network protocol developments.

In Australia, "packet radio is permitted in the Amateur Service", subject to several conditions. Number three on the list is the following:

(3) EACH "PACKET" shall contain the originating stations identification, that of the destination station and the station transmitting (if different from the originating station). [emphasis mine]

and a note

(A) Any protocol may be used for "packet" transmission provided that the identification requirements stipulated in (3) above.

NET/ROM immediately runs into trouble here. The usage of the user's callsign on the "downlink" (see the NET/ROM manual for definitions of these terms) is a clear violation because only the originating station is identified, not the transmitting station. (Some have also argued that this is likely to mislead an observer who is unfamiliar with NET/ROM into thinking that the originating and transmitting stations are one and the same. This has also caused a fair bit of concern in the USA).

Another look reveals that "uplink" packets identify only the originating station and the first NET/ROM node in the path, not the ultimate destination. This may also violate the rules, depending on whether by "destination"

the DoC meant the next hop, the ultimate destination, or both.

Still worse, even packets flowing on a "crosslink" (i.e., between two NET/ROM nodes) may not satisfy the rules. Even though they are encapsulated in NET/ROM's internal network datagram protocol, the addresses in that layer identify the "edge nodes" of the NET/ROM network serving the end users, not the end users themselves.

At this point some may object to my analysis by saying "but aren't the origination and ultimate destination call signs sent across the network when the NET/ROM path is set up, and doesn't this satisfy the identification requirements?" Yes and no. Yes, the calls are sent once at the beginning of a connection, but no, there's strong precedent to say that's not enough. Quoting again from the DoC letter:

Recognising that version "V2" of the Vancouver packet protocol can not meet the identification requirements stipulated until an updated version is released, the Department is prepared to authorise use of "V2" until 31 March 1987. It is anticipated that version "V3" will be available by this time and it is understood that "V3" will fully comply with the identification requirements."

V2's problem is that full call signs are sent only at the beginning and end of communications — not in every packet as stipulated in requirement #3. Ordinary data packets in V2 carry only shortened, 16 bit "IDs" computed by a CRC "hash function" on the call signs. Hashing is inherently a "many-to-one" mapping; otherwise the ID would be the same size as the call sign and there'd be no point in using it. However, this makes it impossible to turn a hashed ID back into a unique call sign; hence the DoC's objection.

I draw several clear implications from these rules and the unfortunate experiences of V2 and NET/ROM:

1. The requirement to identify the originating, transmitting and destination stations in every data packet is tantamount to requiring datagram-style protocols at both the link and network layers.

Networking approaches based on pure virtual circuits at the network layer would therefore not meet the

identification rules. In particular, in the COSI proposal to run the X.25 Packet layer (level 3) protocol above AX.25 Level 2, the originating and ultimate destination stations are fully identified only at virtual circuit setup. The Logical Channel ID carried in each data packet would be insufficient end-to-end identification for the same reason V2's hashed IDs are insufficient link level identification. While it is conceivable that these virtual circuit protocols could be "hacked" to include the required identification in every packet, this would defeat the sole advantage of these protocols vis-a-vis datagram protocols — avoiding the overhead of carrying addresses in every packet!

2. The rules do "not" preclude building virtual circuits (if desired) atop the required datagram protocols. For example, one might want hop-by-hop acknowledgements at the link level. AX.25 is okay here because it really has two sub-layers: an upper connection-oriented LAPB sublayer and a lower datagram address sublayer. However, within a store-and-forward network link level virtual circuits could be used only to carry network-layer datagrams. They could not be concatenated in a connection-oriented "patch panel" or any other scheme that would violate the end-to-end significance of the addresses in the datagram network layer. If the end user requires a virtual circuit, he would have to build it himself by running a true transport protocol on an end-to-end basis on top of the network datagram protocol.

This is how NET/ROM got into trouble. Although it uses datagrams internally, the desire to support "level 2 only" users means they are not extended all the way to the end users. However, if users ran the NET/ROM protocols in their own stations on a fully end-to-end basis, "uplinks" and "downlinks" would disappear. AX.25 level 2 addresses would always indicate actual transmitting and receiving stations, the NET/ROM layer 3 protocol would identify the true originator and the ultimate destination of each packet, and NET/ROM's layer 4 protocol would become a true transport protocol. It seems reasonable to conclude that this would satisfy the Australian rules.

3. It is unknown whether the ARPA IP protocol as it stands would be acceptable under the Australian rules. The question is whether the IP

addresses in each datagram constitute acceptable "identification". For many reasons, IP addresses do not include call signs. However, because IP addresses are assigned on a relatively static, unambiguous basis to individual amateurs with the assignments made publicly available, mapping an IP address back to an individual amateur is relatively straightforward. If required, however, a special "IP option" carrying the necessary call signs could be easily created and added to each datagram. This may be a useful feature even in the US, where it could be added to each datagram entering from outside to indicate the station responsible for clearing its entrance into the amateur network. Of course, a clearly acceptable alternative would be to run IP on top of NET/ROM's datagram network layer.

In short, it appears that Australia's identification rules all but require that an amateur store-and-forward packet network use a datagram protocol at the network layer and be "spoken to" directly by the end users. Protocol "conversion" gateways (as distinguished from protocol "encapsulation" gateways) would not satisfy the requirements.

Letter to the Editor

Glad PSR is Back!

Nice job Scott. PRM was nice but just didn't seem to have the info that the PSR had. It was nice getting PRM every month but I'm glad to see PSR back in my mailbox...even if it is quarterly. Keep up the good work.

73 John, WA4UMR @ WA4UMR

Note from the Editor

Thanks to all of the contributors to this issue! It "feels good" to get an issue wrapped up with such good content.

In future issues, we're looking forward to having Harold Price, NK6K, back among the regular contributors (you may have noticed the absence of his NK6K>Packet column from 73!).

Please continue to get your PSR material to me when its ready rather than waiting for the publication deadline. You can send it to me at the address on the cover of this issue.

- Scott, W3VS

TAPR Board of Directors Nominations

by Lyle Johnson, WA7GXD

Tucson Amateur Packet Radio is a non-profit corporation licensed in the State of Arizona as a scientific and educational institution, and likewise recognized by the IRS as a 501(c)3 tax-exempt organization for these same purposes.

TAPR is run by a Board of Directors. There are fifteen members of the Board (we're bigger than General Motors!), each of whom serves a three year term. To keep some measure of stability to the organization, only five Directors are elected every year. Current Board members and the expiration of their terms are:

Mike Brock, WB6HHV	2/88 +
Tom Clark, W3IWI	2/90
Andy Freeborn, NØCCZ	2/88 +
Steve Goode, K9NG	2/89
Bob Gregory, KB6QH	2/90
Eric Gustafson, N7CL	2/89
Skip Hansen, WB6YMH	2/88 +
Lyle Johnson, WA7GXD	2/89
Scott Loftesness, W3VS	2/89
Dianne Marshall, AL7FG	2/90
Bob McGwier, N4HY	2/89
Dan Morrison, KV7B	2/88 +
Harold Price, NK6K	2/90
Bill Reed, WDØETZ	2/88 +
Dave Toth, VE3GYQ	2/90

In order to nominate someone (including yourself, we're not adverse to that sort of thing, you know) simply send a letter to the TAPR PO Box (NOT VIA PACKET RADIO — THIS IS OFFICIAL BUSINESS) stating your candidate's name, callsign (if any), and qualifications. Include a phone number and mailing address if you can so we can get in touch with the lucky person and see if they want to run.

What is expected of a Board member? Come to the annual meeting and Board meeting in Tucson every year on his/her own nickel. Participate in the decision making process. Provide TAPR officers with guidance during their term. And be a current TAPR member!

In short, have a voice in what makes TAPR work and share in the responsibility (and reward) for what TAPR does.

The pay is the same as I get for being

TAPR president - no money and a few angry letters!

The deadline for nominations is 15 December 1987. This gives us time to contact the candidate and get a statement from them for publication in the January PSR, along with getting their name on the ballot for that same issue.

Letter to the Editor

Looking for Some Tips!

Scott,

I just got the latest issue of PSR, and really enjoyed it! I have been rather out of touch for a year due to major career changes, and am trying to get up to speed. A lot has changed!

However, one suggestion is for a directory of the sources of info/software. In the old days, it was all on DRNet, but that was a closed operation for the inner circle. Now, everything is spread all over the place, or so it seems. For instance, there is CompuServe, the old ARPANet, USENet, and something called UUCP. Obviously, I have an account on CIS, have in the past (10 years ago, when I was a grad student) had an account on ARPANet, and for that matter the Department of Energy's various fusion networks, and am currently on the NASA SPAN system, but I have no idea what the last two are.

What brought this on is the TCP/IP article by N3EUA. Enough info is there, if you read very, VERY carefully, to use the systems blindly to get what you need, but I and two ham-scientist friends I asked to read it came away scratching our heads. These academic and government networks are great toys (I use a couple myself with NASA), but I suspect that some of your less sophisticated supporters could be put off by the cryptic style of information presentation. On the other hand, folks like myself, while not wanting or needing a manual for the service discussed, would like a little more info before using it blindly.

How about an article entitled something like: "A Beginners Guide to Electronic Sources of Technical Packet Information: What They Are, and How to Use Them." Nothing fancy, just a list of the name of the service, a couple of lines on what it is, how somebody who is not a real user

can access it, whether it can be accessed by Telenet, CIS, or other leased phone system, and where to look for info once you get in, and a smattering of the command syntax. For instance, UNIX is sufficiently different from DOS that a "naive" user would be very unlikely to get anywhere on encountering a UNIX system for the first time. (I know I threw up my hands in disgust and went back to my DEC VMS VAX!!!) I would think that somebody who is already on most of them could knock out a quick orientation primer in a few hours, and then find an "outsider" to look it over for the usual assumptions.

73s,

Alan
WA4SCA

Software Offerings from the TAPR Office

Diskettes: All diskette software is IBM PC DOS 2.0 or later format, on 5-1/4" diskettes. Please send a return mailer with postage and \$2 for copying. If you desire TAPR to provide the diskettes, the charge is \$1 for mailer and postage, then add \$0.75 per diskette.

- o WORLI/VE3GYQ C BBS (1 diskette)
- o TCP/IP (KA9Q) (3 diskettes)
- o Intro to TCP/IP (2 diskettes)
- o TNC 1 Source Code (1 diskette)

EPROM:

- o TNC 2 release 1.1.4 (27C256)
- o TNC 2 WA8DED release (27C256)
- o TNC 1 WA8DED release (2 x 2764)
- o TNC 1 KISS (2764)
- o TNC 2 KISS (27C256)
- o TNC 2 1.1.4 w/loader (27C256)
- o TNC 2 1.1.4 w/KISS (27C256)

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

TNC 2 release 1.1.4 requires 32k RAM in your TNC 2. If you haven't already upgraded, 32k RAM chips are available from TAPR for \$20 postpaid. Current blank EPROM prices are \$10 for 27C256 and \$5 for 2764 (may be 27C64, depends on availability).

TNC 2 1.1.4 includes documentation update.

Continued on page 13

AMNET: An Amateur Packet Switched Network

by Paul Flaherty, N9FZX
Project OSCAR

Abstract: The AMNET, a public service packet switched network, is detailed. The network utilizes a one megahertz wide linear transponder aboard the AMSTAR Phase IV satellite, as well as several terrestrial gateway stations. It provides functionality similar to the ARPANET(1) in facilitating long distance network traffic, using the DDN standard Internet Protocol (IP), and Transmission Control Protocol (TCP). The network fulfills the goals of the Digital Community Access System (dCAS) in providing a number of standard functions, including mail transfer (SMTP), bulletin transfer (NNTP), remote file transfer (FTP), and remote login (TELNET) capabilities.

1. Motivation

The motivation for a public service community access system can be found in several papers proposing experiments with the AMSAT Phase III satellites. These early gateway experiments demonstrated both the feasibility and desirability of long distance satellite links for voice repeaters.

With the advent of the AMSAT Phase IV project, several proposals for a simple, easy to access service were motivated by the following:

- 1) To create a reliable long distance emergency communications system that was both easy to use, and inexpensive.
- 2) To involve more amateurs in Amateur Space Program.
- 3) To experiment with new and different concepts for network planning, management, and use.

The voice Community Access System (vCAS) is currently being developed by Jim Eagleson, WB6JNN, of Project Oscar. The current design multiplexes (using FDM) several voice channels onto a wideband linear transponder. The voice channels use a modulation technique known as Amplitude Companded SideBand, or ACSB, which provides FM quality voice in a fraction of the

bandwidth.

Use of the vCAS system involves setting up a connection between a member repeater and a regional gateway, then between two gateways, and finally to the destination repeater. In practice, this would require the entry of a security access code, and then a three digit destination code. One destination code (999) is reserved for a special "CQ" function, which connects the user with a randomly chosen repeater.

The signalling scheme for vCAS has yet to be well defined. However, the current proposal calls for an architecture similar to the AT&T Co - Channel Interoffice Signalling (CCIS) system, with all trunk connection functions performed out of the voice channel.

2. Digital CAS

The Digital Community Access System (dCAS) has been motivated by similar warrants. In particular, the overwhelming growth of interest in Packet Radio over the last few years indicates the popularity of digital communications in the Amateur Community. Also, there is a need for a companion service to vCAS, to implement intergateway signalling.

One of the first dCAS proposals came from the author (3). It involved the time division of a 128 kbps data stream, which was managed by the satellite itself. Channel allocation was obtained by requesting a slot from the satellite on a separate multiaccess channel.

This scheme had several downfalls, including:

- 1) It required the development of a relatively high speed modem, which would be expensive.
- 2) It required the use of a highly stable time base.
- 3) Stations who were too remote from the terrestrial gateways could not use the system.
- 4) It lacked the desired capacity.
- 5) It used circuit switching, which is relatively inefficient when compared to packet switching.

The current dCAS design is much simpler, more accessible, and less

expensive than the first. It involves the use of a frequency division multiplex scheme similar to the vCAS system, but using much wider (20 KHz) channels. Each trunk uses a four - ary QPSK scheme, and digital coding to provide an end user data rate of 9600 bps, and requiring a 12 db signal to noise (S/N) ration (3db margin) for a 1×10^{-6} bit error rate (BER). Using this scheme, about fifty trunks can be provided in the transponder space available. The trunks would then be allocated between the terrestrial gateways, using empirical data to predict relative traffic levels.

This design is in essence a low speed copy of the architecture of the ARPANET. Because of the similarity, all of the research on increasing the capacity of the ARPANET will be applicable to the AMNET; thus dCAS can build on nearly twenty years of experience and exhaustive research. In addition, operation of the AMNET could provide valuable insight into future options for the ARPANET.

This architecture requires the use of the DDN standard protocol suite, including TCP/IP and several other service protocols. The use of TCP/IP does mandate a certain level of sophistication, both on the part of the end user and his equipment. However, any network of this magnitude should require a minimum level of hardware commensurate to its capabilities. In particular, many hobbyist computers lack the capability to deal with higher data rates and multiprogramming capability. A network designed for such computers would be far less efficient, and would lack the desired capabilities. Indeed, a similar comparison would be the now famous battle between AM and SSB on the HF bands during the infant years of Amateur Radio.

For users who are not within distance of a gateway station, remote access would be provided by allocating a few of the trunks as "open ended". A remote user would access the network by choosing an unused, open trunk, and then communicating with one of the gateways via the satellite transponder.

3. User Services

The services provided by dCAS closely follow the services provided by the DDN. In particular:

1) Electronic Mail. The ability to send a message to a particular person has proven to enormously popular in the Amateur Packet Radio community. This capability is provided by the Simple Message Transfer Protocol, or SMTP.

2) Electronic News. Another popular feature of the current packet radio setup is the passing of bulletins, articles, and for - sale messages. The Network News Transfer Protocol, NNTP, provides this service.

3) File Transfer. The ability to transfer software and data files is an important one, especially in a hobbyist community. FTP, the File Transfer Protocol, gives the user the ability to upload and download files from any machine in the network.

4) Remote Login. Another important service is to allow other people in the network to access a computing resource. In particular, persons or groups who own timesharing systems, and are willing to allow public access to them, can present their systems for public use via the network. TELNET, a protocol designed for use between a diversity of terminal and computing devices, provides this capability.

Several other DDN protocols exist, providing secondary services, public information, and other functions which may find use in the AMNET.

4. Summary

The current implementation of dCAS, also known as the AMNET, will provide a high level of functionality to users throughout the footprint coverage of the AMSTAR Phase IV satellite. The network builds upon the experience of DARPA's DDN (also known as the ARPANET), and uses inexpensive components to achieve high reliability at low cost. It allows remote access by stations far removed from metropolitan areas. And, it will allow many popular features of the current terrestrial packet radio network to be extended nationwide.

Paul Flaherty, N9FZX
Computer Systems Lab — Stanford
ARPA: paulf@shasta.Stanford.EDU
UUCP: shasta!n9fzx!paulf@umunhum

TAPR Needs YOUR Support!

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

by Mike Busch, W6IXU

NET/ROM version 1.3 released 1 October 1987

Version 1.3 adds four new enhancements to NET/ROM, including the capability that has been most asked-for since its inception: a CQ capability:

1. There is a new CQ command that implements a sophisticated facility for calling CQ and replying to CQs. This new capability is described in detail in the documentation addendum below.

2. NET/ROM now employs an exponentially-distributed random keyup delay scheme ("p-persistent CSMA") in place of the fixed DWAIT delay used in previous versions. The old DWAIT parameter (PARM #16) has been replaced by two new parameters called P-persistence (PARM #16A) and Slot Time (PARM #16B).

3. Station ID broadcasts can now be made conditional on node activity. The Station ID on/off parameter (PARM #24) has been changed to a three-way switch (values 0, 1, or 2). A setting of 2 (the default) results in ID broadcasts every 10 minutes. A setting of 1 results in ID broadcasts every 10 minutes ONLY IF the node has transmitted since the last station ID broadcast. A setting of 0 disables the station ID broadcasts altogether.

4. NET/ROM now maintains its destination list in sorted order. It uses a sort key composed of the identifier, callsign, and SSID (with the identifier being most significant). This results in a sorted NODES display, which makes it much easier to find a particular node.

The following additions have been made to the NET/ROM manual. Following page 38:

CQ Command

The CQ command is used to broadcast a short text message from a node, and to enable other user stations that receive the broadcast to connect to the station that originated the broadcast. The command is:

CQ [textmessage]

where textmessage is optional and can be any string up to 77 characters long (blanks and punctuation are al-

lowed). Note that the CQ command cannot be abbreviated, since "C" is interpreted as a CONNECT command.

In response to a CQ command, the node broadcasts the specified text message in "unproto" mode, using the callsign of the originating user with a translated SSID as the source and "CQ" as the destination. The broadcast is made in the form of an AX.25 UI-frame with a PID of 'F0' hex. For example, if user station W1XYZ connects to a node and issues the command:

CQ Hey, is anybody there?

the node transmits a broadcast that would be monitored by local users as:

W1XYZ-15>CQ: Hey, is anybody there?

After making the broadcast in response to the CQ command, the node "arms" a mechanism to permit other stations to reply to the CQ. A station wishing to reply may do so simply by connecting his TNC to the originating callsign shown in the broadcast (W1XYZ-15 in the example above). A CQ command remains "armed" to accept replies for 15 minutes (see PARM #15), or until the originating user issues another command or disconnects from the node.

Any station attached to the node in command mode may determine if there are any other stations awaiting a reply to a CQ by issuing a USERS command. An "armed" CQ channel appears in the USERS display as:

```
(Circuit, Host, or Uplink) <-->  
CQ(usercalls)
```

The station may reply to such a pending CQ by issuing a CONNECT to the user callsign specified in the CQ(...) portion of the USERS display—it is not necessary for the station to disconnect from the node and reconnect. For example:

```
^C  
cmd: C BWI  
*** Connected to BWI ***  
USERS  
BWI:M3IWI-5) NET/ROM 1.3 (701)  
Uplink(K1HTV-1) <--> CQ(K1HTV-  
14)  
Circuit(LAS:K7WS-1 W1XYZ) <--> CQ(W1XYZ-  
15)  
Uplink(M4HY)  
CONNECT W1XYZ-15  
BWI:M3IWI-5) Connected to W1XYZ  
Hi, Dr. Bob! Thanks for answering my CQ.  
etc.
```

Users of the CQ command are cautioned to be patient in awaiting a

response. Your CQ will remain "armed" for 15 minutes, and will be visible to any user who issues a USERS command at the node during that time. Wait at least five minutes before issuing another CQ—give other stations a chance to reply to your first one!

NOTE: The CQ command was introduced in NET/ROM version 1.3. On a node using an earlier version, you will get the message "Invalid command".

The following changes apply to Appendix C—Parameters:

Parameter 16A: Keyup P-persistence (p=P/256) (default=64, minimum=0, maximum=255) Together with slot time (parameter #16B), defines the exponential delay algorithm used by the node when keying up its transmitter. When the node has something to transmit and the channel is clear, the node generates a random integer in the range 0-255. If the random number is less than or equal to the P-persistence parameter, the node keys up its transmitter immediately. Otherwise, the node delays for one slot time, generates a new random number, and repeats the procedure. The default value of 64 corresponds to a keyup probability value of 0.25.

Parameter 16B: Keyup slot time (10 ms increments) (default=10, minimum=0, maximum=127) Together with P-persistence (parameter #16A), defines the exponential delay algorithm used by the node when keying up its transmitter. The default value of 10 corresponds to a slot time of 100 milliseconds.

[NOTE: The parameters will be renumbered in the next edition of the NET/ROM documentation. NET/ROM 1.3 has 26 parameters.]

Parameter 24: Station ID broadcasts (2=on, 1=conditional, 0=off) (default=2, minimum=0, maximum=2) Defines whether the node broadcasts station-identification beacons. The default setting of 2 causes station identification to be broadcast every 10 minutes unconditionally. A setting of 1 causes station identification to be broadcast only if the node has transmitted since the last beacon. A setting of 0 disables station identification beacons altogether.

Parameter 25: CQ broadcasts (1=on, 0=off) (default=1, minimum=0,

maximum=1) Defines whether or not the node will broadcast AX.25 I-frames in response to the CQ command. Even if such broadcasts are disabled by setting this parameter to zero, the other features of the CQ command continue to operate normally. The default value of 1 causes CQ broadcasts to be enabled.

by Franklin Antonio, N6NKF

AX.25 PROPOSED CHANGES

To: Committee considering AX.25 changes, and other interested parties.
From: Franklin Antonio, N6NKF
Date: 9/30/87

Lyle Johnson wrote a note in October '86 PRM requesting suggested changes in the AX.25 protocol. I wrote some comments then, but never delivered them to anyone. More recently, several people have been discussing an upcoming committee meeting to discuss this same issue. In light of the current interest, I've dusted off my notes, and am mailing them to the people who I believe are on the committee, and a few others who might be interested in same.

I must comment that where the following text mentions present behavior of the TAPR TNC2 or SW-2000 NET/ROM software it is not intended as criticism of the creators of either product. I have high regard for both.

I've used the term "NC" to specify the union of TNCs, NNCs, PS186's etc, meaning simply "device that implements AX.25 level 2 protocol".

1. DUPLICATE PACKET DETECTION FIX

AX.25 ver 2 does not reliably detect duplicate I-frames. I have often received a line of text twice during a QSO over a poor link. Other operators have also reported seeing this problem. In particular, when N2 is exceeded, AX.25 resets the link, losing track of which I-frames have been previously received, and potentially causing a duplicate I-frame to be accepted.

For example... Station 1 sends an I-frame to station 2, with many re-transmissions. Station 2 successfully receives the I-frame, sends an RR to acknowledge, and delivers the I-frame

to the next higher level protocol (possibly human). Station 1 does not receive the RR. Instead, station 1 exceeds N2, and initiates a reset. After the reset, station 2 has forgotten that this I-frame was already received. Station 1 never knew. Because this I-frame is still in station 1's transmit queue, it is transmitted again. Station 2 receives, acknowledges, and delivers the packet a 2nd time to the next higher level.

When N2 is exceeded while in almost any state, AX.25 presently sends a SABM, and goes to S2 (link setup). I would change that to send a DISC and go to S4 (disconnect request). This would simply disconnect a link when the protocol cannot proceed in a reliable fashion.

Alternatively, if a link reset procedure could be devised which would not result in duplicate or missing I-frames, and thereby ensure continued reliable operation, it could be used in place of the existing link reset.

This is, of course, only one of the situations that invokes the link reset procedure. I believe it is the most common. Other situations (ie FRMR, DM, or UA rcvd while in states S5 thru S16), should be evaluated for their potential for causing duplicate or missing I-frames, and action taken accordingly.

2. CONNECTIONS THAT KEEP COMING BACK

There is another symptom, less often seen, which is also a result of the present link reset procedure. Occasionally, a user will connect to a 2nd station over what turns out to be a very bad RF link. After a short time, it is obvious that the connection isn't going to work, and the user gives up, and disconnects. A few minutes later the connect alarm rings, and the user is amazed to find that an automated BBS system appears to be calling him back! The user disconnects, only to find that his connect alarm rings a few minutes later. The BBS has called him back again! No matter what the user does, the BBS keeps calling him back. One time, after this had happened to me 5 times in a row, I gave up, turned off the TNC, and left the house. Clearly this is an undesired behavior. When the user (or a higher level protocol) gives the disconnect command, he has the right to expect that the connection will in fact be ended.

Here's what happens. the user has given the disconnect command, but his NC only sends the DISC frame N2 times then gives up. None of these got thru. The BBS thinks the connection is still valid, and is trying to send an I-frame (probably its long herald). The BBS tries N2 times, then if still unsuccessful, initiates the link reset procedure, and tries sending a SABM up to N2 times. If one of these SABM's is received by the user's NC, it will accept it as a request to initiate a new connection.

It is unfortunate that the same SABM frame is used for two functions (initiation of connections and link reset).

When N2 is exceeded while in almost any state, AX.25 presently sends a SABM, and goes to S2 (link setup). I would change that to send a DISC and go to S4 (disconnect request). This would eliminate the most common cause of the unintended callback behavior

Alternatively, different S-frames could be specified for the two functions (initiation of connections and link reset). This would avoid confusion between the two functions, and thereby eliminate the unintended callback. SABM's would only cause action in state S1, and "RESET"s would cause action only in states other than S1. Unfortunately, I don't see any way that this alternative can be implemented in a compatible fashion.

3. TYPES OF DISCONNECT

3.1 Specify Forced Disconnect Behavior

The present protocol does not specify what action should be taken when an NC is in state S4 (Disconnect Request) and the operator gives a "Local Stop". In this situation, the TAPR TNC2 goes immediately to state S1 (Disconnected), and stops transmitting DISC frames. If the committee believes this is reasonable behavior, it should probably be specified.

Some implementations (notably NET/ROM when used as a level-2 device) do not implement any forced disconnect. An operator cannot instruct NET/ROM to stop sending DISC's.

3.2 Specify Queued Text Behavior

The present spec does not specify what should be done with text which is

queued for transmission when a disconnect occurs.

The TAPR TNC1 and TNC2 this text is immediately dumped out in UI-frames to the UNPROTO address. This behavior might have served a purpose historically (Someone told me it was done to make sure that everything that was put in "came out somewhere"), it now serves no useful purpose, and in fact causes confusion. Many users leave their UNPROTO address set to "CQ". This results in packets addressed to "CQ" transmitted accidentally. To avoid confusion, some operators set their UNPROTO address to "OOPS". We shouldn't be transmitting accidental packets addressed to either "OOPS" or "CQ".

The specification should be altered to specify that queued text is discarded when either the S1 (disconnected) or S4 (disconnect request) state is entered.

NET/ROM implements a deferred disconnect scheme. When the operator requests a disconnect, NET/ROM guarantees that queued text will be delivered prior to the operator-requested disconnect. (A protocol-initiated disconnect due to T3 or N1 may of course cut short the session, and cause the text to be discarded.) I believe they consider the deferred disconnect logic part of their level-3 protocol.

I believe the proposed change is consistent with such deferred disconnect schemes. Deferred disconnect logic at some higher protocol level may defer giving the "Local Stop Command" to AX.25 level 2 until such time as there is no text queued for transmission on the connection.

In other words, I'm not saying that text should be discarded when the operator tells his TNC to disconnect, but text should be discarded when AX.25 L2 considers the link disconnected (S1) or has entered a state from which disconnection is certain (S4).

4. TIMING ON THE RF CHANNEL

There is a considerable body of unspecified timing behavior which needs to be specified, especially on half-duplex channels. One could argue that such information belongs in a level-1 spec instead of a level-2 spec, and possibly be correct. Right

now, this information is specified nowhere. This situation should change. This change could come about by expanding the scope of the present AX.25 specification (which already contains some timing information), or by adding a companion specification for level-1. I refer to "the specification" below, not meaning to imply which specification.

In particular, some of the considerations associated with the common TNC commands AXDELAY, AXHANG, DWAIT, RESPTIME, TXDELAY should be specified, and the presently unmentioned carrier sensing and TXtail delay should be specified.

4.1 When to Transmit

4.1.1 Carrier Sensing

At the present time, there's no spec that says an NC should wait for lack of carrier on a half-duplex channel before transmitting! All existing NCs do this, but what if a new NC were introduced which did not bother to carrier sense? Would we argue that it was in violation of an unwritten rule?

A carrier-sense requirement for half-duplex operation should be inserted into whatever specification eventually specifies behavior at this level.

4.1.2 Xmit Start Randomization

I believe there is general agreement now that the DWAIT concept should be replaced by some mechanism which randomizes the transmission times of all packets.

Phil Karn, and others, have suggested "p-persistence". While this is only one of a seemingly infinite number of ways to accomplish this randomization, I find no fault with the method.

I suggest that "p-persistent" behavior should be required.

4.1.3 Keyup Delay

All existing NCs contain a mechanism to allow the operator to select the keyup delay, "TXDELAY".

The specification should require that TXDELAY be user settable, and should specify that a setting of zero should indicate that the software is to add no intentional delay.

Why specify zero? Efficient high-

speed operation will require that TXDELAY be implemented in hardware. For example... A radio/modem designed specifically for high-speed packet operation could easily keyup in 500 microseconds. (The Microwave Modules transverters being used with the WA4DSY modems keyup in under 1ms after a small mod which simply removes one capacitor, and they weren't designed for packet.) A modem designed specifically for burst mode operation can acquire an incoming packet in under 100 bit times. (under a reasonable set of assumptions, which I won't go into here.) At 1 Mbit/sec, 100 bittimes is only 100 microseconds. So we might want a TXDELAY as short as 600 microseconds! Clearly, intervals this short are difficult to implement in software. It seems most reasonable to implement these short keyup delays in the modem/radio. This will be possible only if the NC software is willing to implement a "zero" TXDELAY.

Unfortunately, the WA4DSY modem does not implement TXDELAY in hardware. For that reason, we have included a hardware TXDELAY timer in the PS186's TAPR-style modem-disconnect interface daughterboard.

4.2 When to Stop Transmitting

The mirror image of TXDELAY occurs at the end of a packet. A typical NC writes the last byte of the packet into its serial controller chip, then starts a delay I call TXtail. At the end of this timeout, the radio keyline is dropped.

The TXtail delay is required because on the order of 42 bits at the tail of the packet can be queued in hardware at the time a typical NC processor writes the last byte to its serial controller chip. Most SCC chips have something like 2 bytes of queuing (which may contain stuffed bits), then must append 2 characters of CRC, and at least one flag character. In addition, modems may introduce additional delay in scrambling/descrambling or forward-error-correction encoding/decoding circuits. If the NC dropped keyline without a TXtail delay, packets could be truncated, and therefore destroyed.

Unfortunately, no existing NC allows the operator to set the TXtail delay! Existing NC's typically don't know the modem data rate, so simply implement a very long TXtail delay compatible with the slowest possible data

rate. The TNC2 software implements a 60 millisecond TXtail. Speeds don't have to get very high before the TXtail is longer than the information-containing portion of the packet! A settable TXtail would solve the problem for all but very high speed operation, where as described above, we believe the timing should be done in hardware.

The specification should require that TXtail delay be operator settable, and that a setting of zero should indicate that the NC software add no intentional delay (ie drop RTS immediately after the final byte of a packet is written to the hardware)

The PS186 modem disconnect interface daughterboard also contains a hardware timer for TXtail.

4.3 Backoff

Phil Karn has suggested binary exponential backoff be added to AX.25. While I believe we need some scheme which would effectively throttle a loaded packet radio channel, and thereby avoid congestion, I am concerned that exponential backoff may not be a good solution.

Exponential backoff has worked well on media such as Ethernet. On Ethernet, collisions are detected. Presence of collisions is good evidence that we should backoff. Ethernet is a wire. It is designed so that packets should never be lost due to noise. (and in practice very seldom are) Unfortunately, on packet radio channels, we have NO MECHANISM for detecting collisions, and packets are often lost due to noise!

Phil suggests we should simply backoff every time the T1 timer times out (in other words on every retransmission).

Because it can't tell packets corrupted by noise from packets corrupted by collisions, binary exponential backoff would greatly decrease the throughput on a marginal RF link, even when there is little traffic, and there are no collisions. I fear this could be a considerable disadvantage.

Perhaps something less aggressive, for example exponential backoff with a factor lower than 2 would provide a reasonable compromise. Perhaps arithmetic rather than exponential backoff?

5. MAXIMUM PACKET SIZE

I would like to see the maximum size of an I-field increased to 1024 bytes. This would allow lower packet processing overhead in high-speed operation. Unfortunately, a significantly larger maximum packet size might pose a considerable burden to the implementer of a small low-speed switch.

Ideally, a spec change would accommodate both situations, and stay compatible with existing AX.25 L2V2 software. Unfortunately, the present protocol does not contain a mechanism for buffer size (ie packet length) negotiation. One possibility would be to specify that the 256 byte I-field limit be observed by default, and larger I-fields could be used only after successful maximum length negotiation. The request for large packet operation would be designed so that the presently specified response of an AX.25L2V2 DXE would result in a compatible 256 byte limit. I have not attempted to work out the details of such a compatible negotiation mechanism.

I am not proposing that long packets be used in all situations. The protocol should allow us to use longer packets to control overhead associated with high speed operation when the characteristics of the RF channel allow.

6. DATA COMPRESSION (AND OTHER NEGOTIABLE ITEMS)

There has been some discussion on CompuServe recently about whether a data compression algorithm should be included in AX.25. I believe that inclusion of such a spec at this time would be premature. There is not yet agreement on whether data compression is appropriate, which data compression algorithms are appropriate, or at which protocol level they should be implemented.

However, we should consider the implications of trying to insert such a capability at a later time. We would probably want the NCs at both ends of a connection to perform a negotiation to determine whether both could handle data compression. This is difficult to do in a downward compatible fashion, as discussed in the previous section. We could set the stage for future downward compatible negotiation mechanisms now, by making a small

change that would create fields to be used for negotiation, and provide a spec now for the negative response.

The present specification requires that S-frames, and UA frames contain no I-field. I propose that an optional I-field be added to SABM and UA frames. The I-frame in SABM would be used to request the use of negotiable features. The I-frame in UA would be used to accept or deny the requests. A zero length I-frame in SABM would imply request of no negotiable features. A zero length I-frame in UA would imply denial of all negotiable features. Content of nonzero length I-fields would not be specified at this time, but would be reserved for future AX.25 versions.

The intent is that individual bits, or small fields would be used to represent each negotiable feature. The first bit in the SABM I-field, for example, might be assigned the meaning "Can you handle extended (1024 byte) frames?". The second bit might be "Can you handle N6NKF- style data compression?", etc. The corresponding bits in the UA-frame I-field would specify "no" with a "0", and "yes" with a "1".

This change would be extremely simple to implement in existing software. I-fields in received SABM's would simply be allowed, but ignored, and UA frames would be generated with no I-field, as before. Some implementations probably already fail to check the length of SABM frames, and therefore already comply!

7. COMMENTS

KA6IQA, WB6HHV, and KB5MU reviewed a draft of these comments. KB5MU felt that item 3.2 was inappropriate because it attempted to specify TNC behavior which is outside the scope of level 2 protocol. WB6HHV disagreed with item 6, preferring to keep negotiation mechanisms out of level 2.

Thank you for your time, 73,
Franklin Antonio, N6NKF
CompuServe Hamnet: [76337,1365]
Internet: QUALCOMM@ISI.A.EDU

PSR is YOUR Journal!

Please share your knowledge and experiences in the exciting world of Packet Radio in the pages of PSR!

by Harry Bluestein, N6TE

Modifying the TAPR Beta TNC for the TAPR PSK Modem

The venerable, ancient Beta TNC can be modified to work with the TAPR PSK Modem. The following directions assume that the TNC has been upgraded to be fully compatible with the TNC 1 software as outlined in the TAPR-provided "Beta Upgrade Kit."

1) Install the 20-pin male header (provided in the PSK modem kit) in the wire-wrap area of the Beta board. I located it parallel to J2, leaving two empty rows of holes between J2 and the new header. Solder pins 3 and 16 of the header to the board to secure the header in place.

2) On the underside of the Beta board, locate the plated-through hole between pins 39 and 2 of the WD1933 HDLC controller (U17). Cut the trace coming from that hole. Solder a jumper wire between that plated-through hole and pin 1 of the 20-pin header.

3) Solder a jumper wire from pin 5 of the XR2211 (U18) to pin 2 of the 20-pin header.

4) Unsolder the drain lead of transistor Q2 (VN10KM) and its 4.7k resistor (5 volt pull up) from their connection to pin 27 of the WD1933 (U17). [That connection may have been made at JP3 or at a plated-through hole under U17, depending on when that Beta board modification was performed.] Solder a jumper wire between the floating junction of the VN10KM drain with its 4.7k resistor and pin 18 of the 20-pin header.

5) Solder a jumper wire from pin 27 of the WD1933 (U17) [use the plated-through hole connected to that pin that is located under U17] to pin 17 of the 20-pin header.

6) Solder a jumper wire from pin 26 of the WD1933 (U17) to pin 12 of the 20-pin header.

7) Solder a jumper wire from pin 1 of the 74LS86 (U31) and pin 19 of the 20-pin header.

8) Solder a jumper wire between the GND bus in the wire-wrap area and pin 15 of the 20-pin header.

9) Install the PSK modem interface board on the 20-pin header with the connecting wire leading away from J2. I cut the hole in the rear wall of my TNC enclosure through which I routed the connecting wire from the interface board.

The Beta board will act like a TNC 1, and the PSK modem should be set up for connection to a TNC 1.

TAPR Software

Continued from page 7

WA8DED software includes printed documentation.

KISS and loader software does not include documentation - refer to the TCP/IP source code and user documentation diskettes.

TAPR would like to extend its thanks and appreciation to Ronald Raikes, WA8DED, for allowing us to provide his TNC code.

If you are wanting to run AX25L2 Version 2.0 (required for FUJ/OSCAR 12), or use a TNC 1 or Beta TNC for a digipeater, etc., then Ron's code is your only option.

TAPR has not yet succeeded in bringing the much ballyhooed version 4.0 to light (don't hold your breath on this one).

If you use the WA8DED host mode, or prefer that all your TNCs have the same user interface, we also offer the TNC 2 version of Ron's code.

Note that Ron has donated this code to Amateur use. He doesn't receive

How to Contribute Articles to PSR!

Articles for PSR are prepared and laid out on a Macintosh Plus using Pagemaker and an Apple Laserwriter Plus printer.

Please contribute articles in plain ASCII format via electronic mail:

CompuServe: 76703,407
AT&T Mail: SLoftesness
MCI Mail: SLoftesness
Packet Radio: W3VS @ AA4RE
UUCP: w3vs@cup.portal.com

or in document form to the editor's address on the cover of this issue.

By Lyle Johnson, WA7GXD

PSK Modem Update

The new manual is at the printers as this is written. If you bought a PSK modem kit and sent in comments, you will get a copy.

If you did, and don't see your new manual by the end of October, contact the office and we'll get one out to you. If you are one of those who built the kit and didn't send in comments — shame, shame! This article may be of more than passing interest to you! There were errors on the schematic, so the complete, revised schematic of the PSK modem is reproduced here. For those of you on the fence who think \$100 is too much for a PSK modem, look over the schematic. There are lots of parts here!

Some excerpts from the current manual are presented here for radio configuration and interfacing.

NOTE: We are still looking for inputs from anyone who has interfaced the PSK modem to any other kind of TNC than the TAPR TNC 1, TNC 2, and exact clones. We need inputs on the AEA PK87 and PK232, any of the Kantronics line of TNCs, the PAC-Comm TNC-220, a VADCG board, etc.

See elsewhere in this PSR for interfacing the PSK modem to the venerable TAPR Beta TNC.

RADIO INTERFACING

The information below was gathered from reports sent in by the folks who built the prototype and first kit PSK modems.

Kenwood TS711A/811A

The connections for the Kenwood TS711 and 811 are given below. In addition, these same connections apply if you are using a TS440 or 940 along with a transverter or receiving converter. Except for the 13-pin DIN connector option, the same connections also apply to the TS430 and a transverter or receiving converter. Other Kenwood radios, such as the TR9000 and TR9100, use the same JMP5/6 configuration.

J3		8-pin MIC	Ext Spkr	13-pin DIN
1	Tx Audio	1	or	11
2	Ground	7	Shell	or 4,8,12
3	PTT	8		or 13
4	Rx Audio		Tip	or 3
5	RF DCD	(n/a)		

J4				
1	AFC Common	7		
2	Ground	7	Shell	
3	AFC Down	3		
4	Rx Audio		Tip	
5	AFC Up	4		

JMP5

o o

|

o o

|

o o

o o

|

o o

|

o o

JMP6

Yaesu FT726R

The same JMP5/6 connections apply to other Yaesu models such as the FT790R.

J3		8-pin Mic	Ext Spkr
1	Tx Audio	8	
2	Ground	7	Shell
3	PTT	6	
4	Rx Audio		Tip
5	RF DCD	(n/a)	

J4			
1	AFC Common		
2	Ground		
3	AFC Down	3	
4	Rx Audio		
5	AFC Up	1	

JMP5

o o

o o

o o

o o

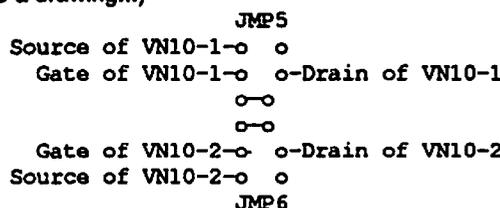
o o

o o

JMP6

ICOM Radios

Some ICOM models incorporate amplified microphones. These radios will typically require more drive than usual. ICOM radios have an odd UP/DOWN arrangement, utilizing a 470 ohm resistor between the UP and DOWN inputs. In addition, they may require a better "ground contact" than that provided by the optoisolators. In these cases, a FET such as a VN10 may be used, driven by the outputs of U12, with JMP5 and 6 wired as shown below. (The real manual has a drawing...)



In addition, tie a resistor of 10k or so from the gate to the source of each VN10. Finally, tie a 470 ohm resistor from the drain of VN10-1 to the drain of VN10-2.

NOTE: This is only a suggested circuit and has not been verified! Finally, note that if the microphone is left connected, the 470 ohm resistor may already be in place through the microphone connections. If you elect to put the resistor inside the PSK modem, you may have to disconnect the microphone.

by Roy Engehausen, AA4RE

In the Mailbox

Not enough people complained last issue about my writings so I got stuck again. Please either complain about this stuff or send me more and better things to write about. I would appreciate any feedback.

Latest Software/Hardware

WORLD has released version 4.0 since my last writing. New features are compatibility with the WA7MBL reverse forwarding, automatic deletion of old messages (as you desire), and the ability to "hold" messages until a SYSOP can review them. The read command also suppresses all the routing headers unless requested.

WA7MBL version 4.0 is in test at W3IWI and K7PYK but is not in general distribution yet.

JDR Microdevices in San Jose still has the 4 serial port cards for \$79.95. The JDR part number is MCT-MS. The cards can be used to add up to 4 serial ports to your mailbox system. Properly modified, all 4 ports can share an interrupt line. Contact JDR at (800)-538-5000.

High Availability Mailbox

Soon after we added a 220 Mhz port to the local mailbox, the user complaints of always being busy increased dramatically. The new port made it possible for the BBS to forward more effectively and traffic increased. Most mailboxes were spending their time forwarding. Fortunately, a solution was already available.

Many people use a multitasking program such as DoubleDos to allow

them to use their computer for a mailbox and to also be able to run their own programs at the same time. Why not run multiple copies of the mailbox in the same computer? Hank, W0RLI, developed an in memory pipe which allows two copies of his "C" BBS program to connect to each other without hardware. Then they can forward between the different copies as if they were on different computers.

This is in production on many mailboxes and the users love it. Now they can connect on two meters regardless of what the 220 port is doing. Its not a very elegant solution to the problem but it certainly helps. High usage mailboxes with multiple ports might want to investigate this idea.

KA2BQE version 95c allows the copies of the program to share the same mail and user files eliminating the need for the in-memory pipe. Look for this idea to spread to W0RLI soon.

Host Mode

For several years now, old TAPR TNC-1 owners have known about and used the WA8DED code for their boxes. This code supports AX25 V2L2 while the TAPR ROMs didn't. There was another feature of the code which was never exploited fully called "HOST MODE." In host mode, the messages between the TNC and the computer are in a format which makes writing programs (such as a mailbox) a heck of a lot easier. Channel numbers and packet lengths are attached to the data thus saving the programmer from having to figure how where one packet ends and another starts.

WA8DED Host Mode was limited to the TNC1 which inhibited use of it. However, TNC-2 and PK-87 support is now available. Two mailboxes are known to be using host mode: W6IXU and WD6CMU. Both programs feature multiport and multiconnect per port.

Another host mode has recently been announced for the TNC-2 or PK-87 running the NETROM software. I have seen a copy of the protocol and it looks very similar to the old WA8DED system but has some improvements. Software 2000, the company that makes NETROM, has distributed test copies of the software to both W0RLI and WA7MBL for

possible inclusion in their mailbox programs.

Zip Code versus Area Code

The NTS/Packet people in several states have adopted the zip code as the routing indicator for third party traffic. Indiana and several east coast states are using area codes and California will soon follow. The only "minor" detail to follow is whether to use five digits or three.

Now that a direction has been established, let's all help implement the idea so the we can stop spending so much time tailoring our forwarding files.

Food for Thought — One Man's Opinion

Competition has been with packet radio a long time. I think it all started with the VADCG versus AX25 controversy. We now have all sorts of things bouncing around in the packet world. Zip code versus area code; NETROM versus COSI versus TCP/IP versus TEXNET; W0RLI CBBS versus WA7MBL versus KA2BQE mailboxes; etc, etc.

Competition is good. Unfortunately, the real world does not have the nice black and white decisions we would like. Each competing system has advantages and disadvantages that have to carefully examined, and, in most cases, tried. Opinions have to be heard and the facts weighed.

The result is usually a good (if not the right) decision. Most packeteers can agree today that AX25 was the better way to go so the system does work.

The trick to competition is timing. At some point, a majority of packeteers will be going in the same direction. The proponents of the other alternatives have a choice: they can either abandon their approach and help improve packet for us all or they can keep sapping the strength of amateur radio by spending resources and energy on a "dead end."

If you find yourself going North while everyone else is going South, it may be time to say "Ah, What the Heck" and join the crowd.

73, Roy AA4RE
AA4RE @ AA4RE

By Dr. Bob McGwier, N4HY

Update on TAPR/ AMSAT Joint Digital Signal Processing Project

The TAPR/AMSAT DSP project is moving forward very rapidly with many interesting developments.

First the software. In the last article one of the things mentioned was that we would like to have a program that takes PSK and turns it into the FSK that a TNC-2 can read. This was accomplished in a program running on a TMS32010. Every time the program sees a binary 1 it outputs a 1200 Hz tone. Every time the program dedodulates a PSK binary 0, it outputs a 2200Hz tone. Given that we run this on JAS-1 signals or the PSK produced from the TAPR PSK modem kit, we can decode the packets on an UNMODIFIED tnc using this technique. N4HY completed this code and tested it on JAS-1 PSK downlink with success. We do not have provision at this time for controlling the frequency of the receiver with the board but could be added with some trouble in the future.

The DSP project is picking up new folks every day. Ten boards were ordered in the first round and it looks like the second round will have to be in the ten to fifteen range. David Langmann, owner of Delanco Spry says delivery will occur on the first batch before the third week of October. We have around 50 official enquiries to both Tom and I. 12 folks have written checks for \$525 for the boards. Some of the enquiries have come from people who have the board already so the total number is higher than those ordered in this first run of boards.

N4HY did a DSP demonstration at the Midwest ARRL convention as a guest of the Central Iowa Technical Society and was very well received. Many new folks and old friends showed up and signed up as interested in the project. Demonstrated were the spectrum analyzer code that Tom W3IWI and Bob N4HY have worked on for some time, Voice digitization and processing were demonstrated and the diversity of possible filters done in software were also shown. Thanks for the hospitality!

During the third weekend of October, W3IWI will be on a major EME outing at the Greenbank National Radio Astronomy Observatory. With a 140' dish, he will be helped by several in running a very fun outing for EME'ers with activities supported on 432 Mhz, 1.2 Ghz, 2.4 Ghz, and 10 Ghz. Also, DSP experiments will be tried between W3IWI and N4HY using the OSCAR class station at N4HY and the use of digital signal processing to enable a QSO which would otherwise probably be impossible. These DSP experiments will occur on the 432 Mhz band and we will be using the TMS32010 boards acquired from Delanco Spry and being made available to members of the DSP project.

On the future of DSP hardware, the folks in Tuscon, under the able leadership of Lyle WA7GXD with Dan Morrison, Mike (Dan's boss N7??), Eric Gustafson N7CL, Chuck N0ADI, and (ask Lyle) have been roughing out ideas for a board for the DSP future of ham radio. Heated bits have been flying back and forth on HamNet and work towards a DSP engine designed for ham radio needs is rapidly converging. With Tom and I saying what we would like in the way of processor capability and the group there including some very able digital signal processors working on the practical implementation of all these competing interests, we are about to converge on a design. It appears that unless the earth shakes in New Jersey, Maryland, and Tuscon we will have a TMS320C25 running at 20 Mhz with a 40 Mhz option for those who can afford the very expensive memory or can't afford to do without the speed. A host processor on the board and probably of the iApx86 family to handle communications between your host computer and the DSP chip.

This will allow folks who don't have PC's to also participate in the fruits of the DSP labor. It will slow down the transfer of data between the host and the DSP chip, but will open the array of DSP tools to folks using anything from Vic-20's to Sun 3's. The applications being worked on are TNC-3's, RTTY, AMTOR demodulators, Telephone modems, spectrum analyzers, SSTV demodulators, FAX demodulators including WEFAX APT, and others that I am forgetting. The number of jobs this engine can do, which will fit in a box not unlike a TNC-2, are limited only by the number of hours in a day that the people working on the project can afford to put in on it.

When the higher speed parts are included and a very fast connection is made to your computer (SCSI port or something similar), it is hard to envision many special purpose devices we currently use in a signal processing way in our shack that will not be replaced by this device. Again the many many things this will replace will be done by simply changing the software running in the DSP chip and its host processor. This project is getting more exciting every day! Be sure to stay tuned for further developments!

How to Subscribe to PSR!

The only way you can subscribe to TAPR's *Packet Status Register* is by being a member of TAPR.

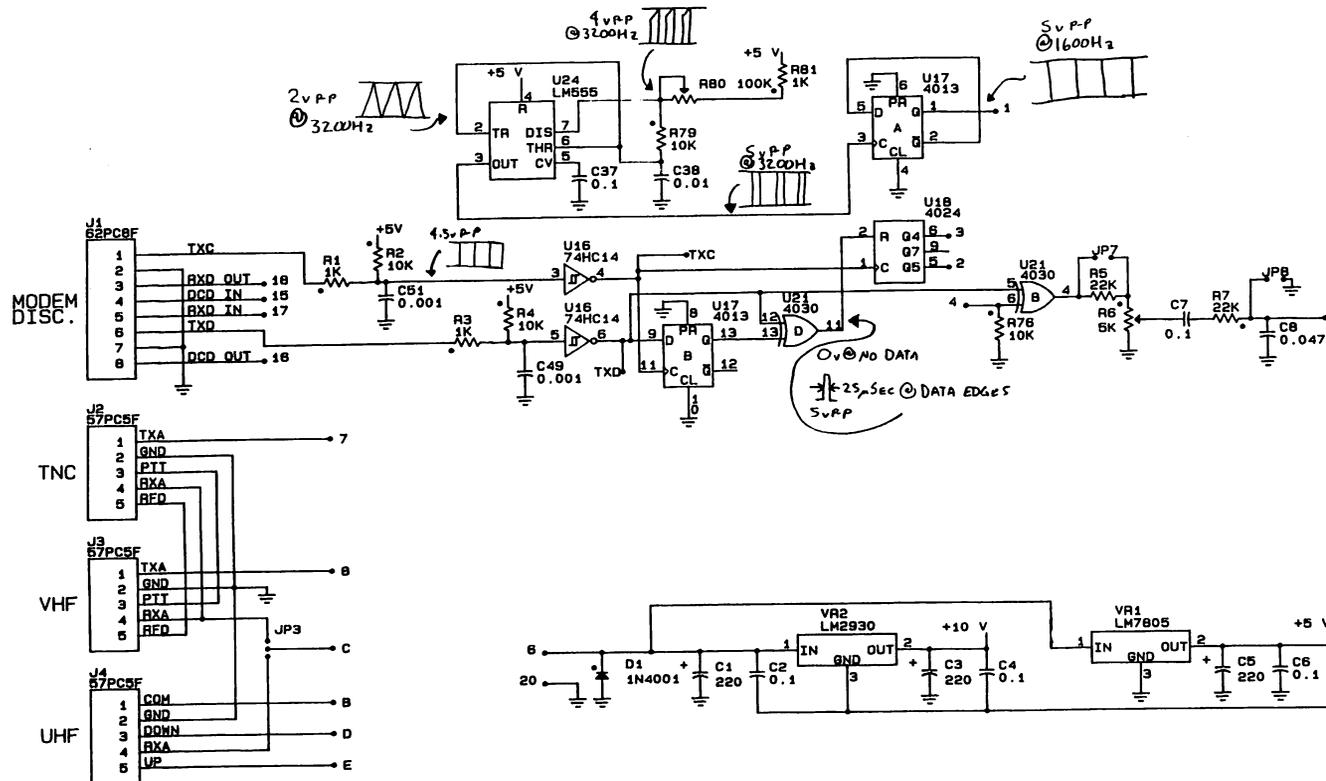
You'll find a handy membership application/renewal form on the back cover of this issue.

Please use it to return your membership renewal of a new application for membership.

Packet Status Register is YOUR journal. With the unfortunate demise of *Packet Radio Magazine* (which was not financially affiliated with TAPR in any way), PSR is now the only source of regular updates on what's happening in the world of Packet Radio.

So, please keep your subscription current my keeping your membership current.

By the way, back issues of *PSR* and *Packet Radio Magazine* (those issues that included PSR) are available from the TAPR office at the address/ telephone number listed on the cover. If you're missing any issues, you can get them replaced by purchase from the TAPR office.



MAIN PC BOARD (1 OF 2)

POWER DISTRIBUTION

PART	TYPE	GND	+5V	+10V
U1, 8	CD4066B	7	14	
U2, 3, 7, 8	TLO84	11	4	
U10	TLC274	11	4	
U4	74C164	7	14	
U5, 18	CD4024B	7	14	
U6	CD4046B	8	18	
U11	LM555	7	14	
U16	74HC14	7	14	
U17	CD4013B	7	14	
U21	CD4030	7	14	
U24	LM555	1	8	

BYPASSES

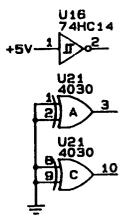
C10-12
C39-40
C42-47
C50

LAST USED

C51
D3
J4
JP8
R82
U24
VR2

UNUSED

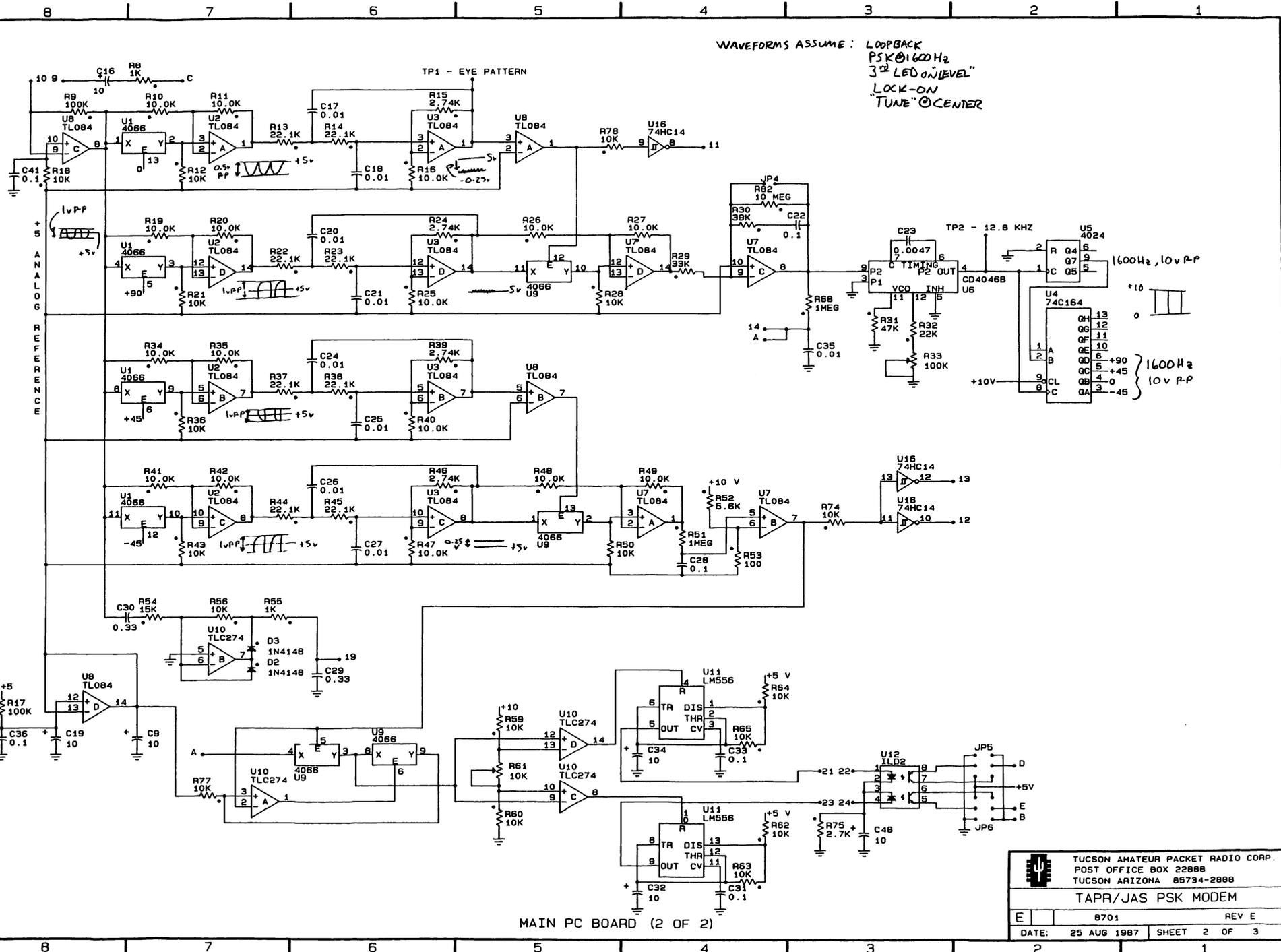
C13-15
JP1-2
R57, 58, 66, 67, 69-73
U13-15, 19, 20, 22, 23



- NUMBERED PAD FUNCTIONS**
- PSK 1600 HZ CLOCK SIGNAL OUTPUT (TTL)
 - JAS 1200 HZ CLOCK SIGNAL OUTPUT (TTL - TNC 1)
 - JAS 1200 HZ CLOCK SIGNAL OUTPUT (TTL - TNC 2)
 - PSK/JAS CLOCK SIGNAL INPUT (TTL)
 - XMIT AUDIO SIGNAL FROM PSK MODULATOR
 - + 11-16 VOLTS DC POWER IN
 - XMIT AUDIO FROM TNC FSK MODULATOR
 - XMIT AUDIO SIGNAL TO VHF RADIO
 - AUDIO SIGNAL OUT (TO LEVEL POT)
 - AUDIO SIGNAL IN (FROM LEVEL POT)
 - AUDIO SIGNAL IN (FROM PSK DEMODULATOR)
 - RCVE DATA (TTL) FROM PSK DEMODULATOR
 - DCD OUTPUT TO DISPLAY BOARD
 - DCD (TTL) FROM PSK DEMODULATOR
 - TUNE SIGNAL TO DISPLAY BOARD
 - DCD (TTL) TO TNC HDLC CONTROLLER
 - DCD (TTL) FROM TNC FSK MODEM
 - RCVE DATA (TTL) TO TNC HDLC CONTROLLER
 - RCVE DATA (TTL) FROM TNC FSK MODEM
 - LEVEL SIGNAL TO DISPLAY BOARD
 - POWER COMMON IN
 - AFC STEP DOWN PULSE TO SWITCH
 - AFC STEP DOWN PULSE FROM SWITCH
 - AFC STEP UP PULSE TO SWITCH
 - AFC STEP UP PULSE FROM SWITCH

- NOTES**
- RESISTORS AND DIODES ARE MARKED WITH A DOT INDICATING THE VERTICAL, OR "STOOD UP." END.
STOOD UP R201 10K AGAINST PC BOARD
 - LETTERED PADS INDICATE SCHEMATIC TIE POINTS, NOT PHYSICAL PADS ON THE PC BOARD. PHYSICAL PADS ARE MARKED WITH NUMBERS OR LEGENDS (EG. +5V).

		TUCSON AMATEUR PACKET RADIO CORP.	
		TUCSON ARIZONA 85734-2888	
TAPR/JAS PSK MODEM			
E	8701	REV E	
DATE:	25 AUG 1987	SHEET	1 OF 3



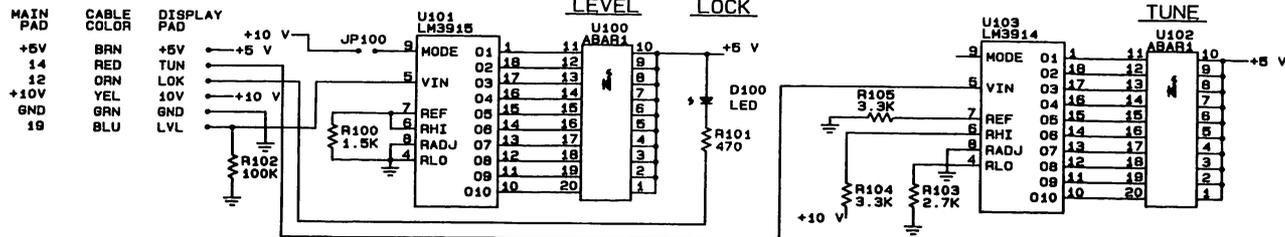
WAVEFORMS ASSUME : LOOPBACK
 PSK @ 1600 Hz
 3rd LED ON LEVEL
 LOCK-ON
 TUNE @ CENTER

ANALOG REFERENCE

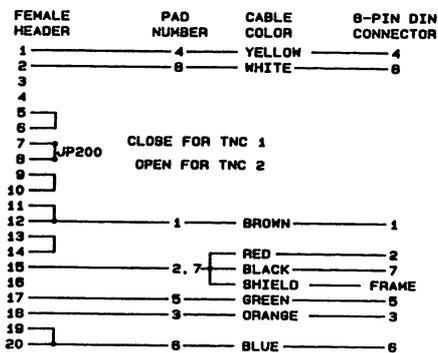
A B C D

MAIN PC BOARD (2 OF 2)

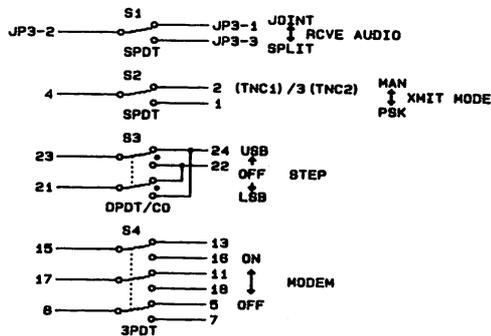
		
TUCSON AMATEUR PACKET RADIO CORP. POST OFFICE BOX 22888 TUCSON ARIZONA 85734-2888		
TAPR/JAS PSK MODEM		
E	8701	REV E
DATE:	25 AUG 1987	SHEET 2 OF 3



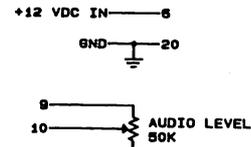
DISPLAY PCB (REV B)



MODEM INTERFACE PCB AND CABLING



FRONT PANEL SWITCHES AND INTERCONNECT



OTHER

POWER DISTRIBUTION

PART	TYPE	GND	+5V	+10V
U101	LM3915	2	3	
U103	LM3914	2	3	

BYPPASSES

C100-102

LAST USED

C102
D100
JP200
R105
U103

UNUSED

JP101-188

LABELLED PAD FUNCTIONS

DISPLAY:
+5V - POWER TO DISPLAY LEDS
TUN - TUNE SIGNAL TO DISPLAY
LOK - LOCK SIGNAL TO DISPLAY
10V - POWER TO BAR GRAPH DRIVERS
GND - POWER AND SIGNAL COMMON
LVL - LEVEL SIGNAL TO DISPLAY

MODEM INTERFACE:

- 1 - TX CLOCK FROM TNC (X16 OR X32)
- 2 - GROUND
- 3 - RX DATA FROM TNC MODEM
- 4 - DCD TO TNC HDLC CONTROLLER
- 5 - RX DATA TO TNC HDLC CONTROLLER
- 6 - TX DATA FROM TNC
- 7 - GROUND
- 8 - DCD FROM TNC MODEM

 TUCSON AMATEUR PACKET RADIO CORP. POST OFFICE BOX 22888 TUCSON ARIZONA 85734-2888		
TAPR/JAS PSK MODEM		
E	8701	REV E
DATE: 25 AUG 1987	SHEET 3	OF 3

MEMBERSHIP APPLICATION

Tucson Amateur Packet Radio Corporation
PO Box 22888, Tucson, AZ 85734

Name: _____

Call License
Sign: _____ Class: _____

Address: _____

City & ZIP
State: _____ Code: _____

Home Work
Phone: _____ Phone: _____

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

I hereby apply for membership in TAPR. I enclose \$15.00 dues for one year's membership dues.

Signature: _____ Date: _____

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

The officers of the Tucson Amateur Packet Radio Corporation are:

Lyle Johnson, WA7GXD	President
Tom Clark, W3IWI	Executive Vice President
Dianne Marshall, AL7FG	Secretary
Terry Price, N6HBB	Treasurer

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

**TAPR Membership &
PSR Subscription Mailing Address:**
Tucson Amateur Packet Radio Corp.
PO Box 22888
Tucson, AZ 85734
(602) 746-1166

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

Packet Status Register - October 1987

**Tucson Amateur Packet Radio Corp.
PO Box 22888
Tucson, AZ 85734**

*Second Class Permit Pending
Tucson, AZ*



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